

日 本 国 特 許 庁
JAPAN PATENT OFFICE

U.S. PTO
09/903683
07/13/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日
Date of Application:

2000年 7月14日

出 願 番 号
Application Number:

特願2000-213855

出 願 人
Applicant(s):

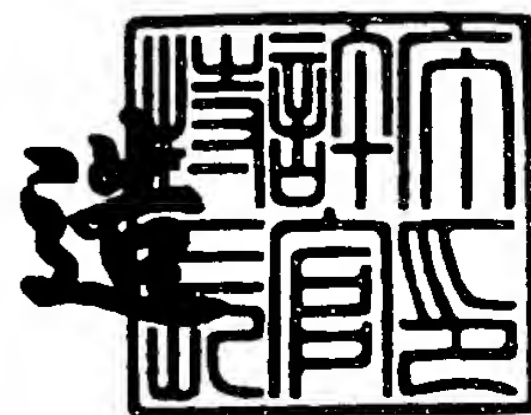
日本電気株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年 5月25日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3043403

書類名】 特許願
【整理番号】 60301671
【提出日】 平成12年 7月14日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 9/06
【発明者】
 【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内
 【氏名】 米山 英晴
【特許出願人】
 【識別番号】 000004237
 【氏名又は名称】 日本電気株式会社
【代理人】
 【識別番号】 100088959
 【弁理士】
 【氏名又は名称】 境 廣巳
【手数料の表示】
 【予納台帳番号】 009715
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9002136
【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ソフトウェアコンポーネント自動生成システム

【特許請求の範囲】

【請求項 1】 パラメータを含む一連のメッセージを受け付けて所定のトランザクションを実行するトランザクション処理プログラムが動作するメインフレームにアクセスするソフトウェアコンポーネント及び該ソフトウェアコンポーネントを利用するクライアントプログラムのソースコードを生成するシステムであって、

前記ソフトウェアコンポーネントを生成するのに必要なモジュール情報、コンポーネント情報、メソッド情報およびパラメータ情報に加えて、前記クライアントプログラムを生成するのに必要となる情報を利用者入出力装置から入力してソフトウェアコンポーネント定義情報を生成するソフトウェアコンポーネント定義部と、

前記ソフトウェアコンポーネント定義情報に基づいて前記ソフトウェアコンポーネントを生成するソフトウェアコンポーネント自動生成部と、

前記ソフトウェアコンポーネント定義情報に基づいて前記クライアントプログラムのソースコードを生成するクライアントプログラム自動生成部とを備え、

前記クライアントプログラムを生成するのに必要な情報として、前記トランザクション処理プログラムに与えるメッセージと 1 対 1 に対応する前記ソフトウェアコンポーネント中のメソッドの呼び出し順の情報を含み、前記クライアントプログラム自動生成部は、前記メソッドの呼び出し順の情報を参照して、クライアントプログラム中のメソッドの呼び出し部分の記述を生成するコンポーネント呼び出し生成手段を含むことを特徴とするソフトウェアコンポーネント自動生成システム。

【請求項 2】 前記クライアントプログラムを生成するのに必要な情報として、更に、前記トランザクション処理プログラムに与えるメッセージで指定するパラメータの入出力種別の情報を含み、前記クライアントプログラム自動生成部は、前記パラメータの入出力種別の情報を参照して、入力となるパラメータを持つメソッドの呼び出し前に当該パラメータへ値を設定する処理を記述する前処理

生成手段を含むことを特徴とする請求項 1 記載のソフトウェアコンポーネント自動生成システム。

【請求項 3】 前記クライアントプログラム自動生成部は、前記パラメータの入出力種別の情報を参照して、出力となるパラメータを持つメソッドの呼び出し後に当該パラメータの値を取得する処理を記述する後処理生成手段を含むことを特徴とする請求項 2 記載のソフトウェアコンポーネント自動生成システム。

【請求項 4】 前記クライアントプログラム自動生成部は、前記コンポーネント呼び出し生成手段、前記前処理生成手段、前記後処理生成手段に加えて更に、クライアントモジュールの骨格を生成するクライアントモジュール生成手段と、入力、出力、入出力として使われるパラメータとして使うための変数の宣言を生成するパラメータ定義生成手段を備えることを特徴とする請求項 3 記載のソフトウェアコンポーネント自動生成システム。

【請求項 5】 前記クライアントプログラム自動生成部は、クライアントプログラム生成ルールに従ってクライアントプログラムの生成を行う請求項 4 記載のソフトウェアコンポーネント自動生成システム。

【請求項 6】 前記クライアントプログラム生成ルールは、前記パラメータ定義生成手段によって参照されるパラメータ定義生成ルール、前記前処理生成手段によって参照される前処理生成ルール、前記後処理生成手段によって参照される後処理生成ルール、前記コンポーネント呼び出し生成手段によって参照されるコンポーネント呼び出し生成ルール、および前記クライアントモジュール生成手段によって参照されるクライアントモジュール生成ルールから構成される請求項 5 記載のソフトウェアコンポーネント自動生成システム。

【発明の詳細な説明】

【 0 0 0 1】

【発明の属する技術分野】

本発明はプログラムの自動生成に関し、より具体的にはメインフレームにアクセスする機能を持つソフトウェアコンポーネント及びこのコンポーネントを利用するクライアントプログラムを自動生成するシステムに関する。

【 0 0 0 2】

【従来の技術】

M i c r o s o f t 社が 1 9 9 6 年に発表した A c t i v e X は、W i n d o w s をベースとしたネットワーク環境で、分散コンピューティングや、ソフトウェアのコンポーネント化の実現を容易にした。それに伴って、最近においては更にコンポーネントの自動生成が試みられるようになってきた。

【0003】

この種の従来技術としては、メインフレームにアクセスする A c t i v e X コンポーネントを生成する M i c r o s o f t 社の C O M T I がある。但し、これはメインフレームにアクセスするためのコンポーネントを作成することはできるが、作成されたコンポーネントを利用するクライアントプログラムの生成を行うことはできない。このため、作成されたコンポーネントの公開情報などを基にクライアントプログラムの開発者が別途にクライアントプログラムを手作業で開発する必要があった。

【0004】

コンポーネントの自動生成技術の他の従来例として、特開平 1 1 - 2 2 4 1 8 4 号公報に記載されたオブジェクト指向データベース操作プログラム生成システム及び方法がある。同公報には、オブジェクト指向データベースに定義されているスキーマ情報を取得し、この取得したスキーマ情報とプログラムテンプレートファイルを基に、オートメーション通信のためのオートメーションサーバ（コードコンポーネント）のためのプログラムを自動生成し、同時に、このオートメーションサーバが公開するデータベースのスキーマクラスとその属性、メソッドを利用する外部アプリケーションとなるクライアントプログラムも自動生成する技術が記載されている。

【0005】

【発明が解決しようとする課題】

作成されたコンポーネントを利用するクライアントプログラムの開発を、コンポーネントの作成と切り離して別途にすべて手作業で行うと、作業工数が増大するなど、クライアントプログラムの開発者に大きな負担がかかる。また、最悪の場合にはクライアントプログラムの開発を進めることができない事態を招く。以

下、メインフレーム上で動作するトランザクション処理プログラムにアクセスする機能を持つソフトウェアコンポーネントとそれを利用するクライアントプログラムの生成を例に、本発明が解決しようとする課題を具体的に説明する。

【 0 0 0 6 】

図 1 9 (a) に示されるように、メインフレーム（例えば N E C 製 A C O S 2 / 4 / 6 ） 3 0 0 上で動作するトランザクション処理プログラム 3 0 1 に対するアクセスは、従来、 E T O S 5 2 G など専用オンライン端末 3 0 2 から行われていた。図 1 9 (b) に、或る 1 つのトランザクション処理プログラム 3 0 1 に対する専用オンライン端末 3 0 2 からの操作手順の例を示す。この例では、メッセージ M 1、メッセージ M 2、メッセージ M 3 の順で、専用オンライン端末 3 0 2 からメインフレーム 3 0 0 上のトランザクション処理プログラム 3 0 1 に対してメッセージを送出することで、一連のシーケンスをトランザクション処理プログラム 3 0 1 に行わせている。メッセージ M 1 ~ M 3 では必要に応じて幾つかのパラメータ P R 1 ~ P R 3 が指定される。個々のパラメータは、・入力パラメータ、・出力パラメータ、・入出力パラメータ、或いは・パラメータとして働かない、の何れかの状態をとる。何れの状態をとるかは、メッセージ M 1 ~ M 3 のメッセージ I D に応じて動的に変化する。

【 0 0 0 7 】

このようなメインフレームのトランザクション処理プログラムに対するアクセス機能を専用オンライン端末からでなく、 W i n d o w s ベースの汎用パーソナルコンピュータから可能にする場合、図 1 9 (c) に示されるように、メッセージ M 1 ~ M 3 に 1 対 1 に対応するメソッド M T 1 ~ M T 3 を実装したソフトウェアコンポーネント 3 0 3 を作成し、また、それを利用するクライアントプログラム 3 0 4 を作成することになる。従来においては、ソフトウェアコンポーネント 3 0 3 はソフトウェアコンポーネントを作成するのに必要な各種の定義情報から自動的に生成し、クライアントプログラム 3 0 4 は生成されたソフトウェアコンポーネントの公開情報などを基にクライアントプログラム開発者が別途に手作業で作成していた。

【 0 0 0 8 】

この場合に問題となるのは、1つは、作成されたソフトウェアコンポーネント 3 0 3 の公開情報を見てもメソッドMT 1 ~ MT 3 をどのような順序で起動すべきか分からないため、クライアントプログラム 3 0 4 の開発者にとってメソッドMT 1 ~ MT 3 の呼び出し順の記述の作成が容易でないことである。メソッドMT 1 ~ MT 3 の起動順序を理解するには、メソッドMT 1 がメッセージM 1 に、メソッドMT 2 がメッセージM 2 に、メソッドMT 3 がメッセージM 3 にそれぞれ対応していること、及びメインフレーム 3 0 0 のトランザクション処理プログラム 3 0 1 の仕様ではメッセージM 1、M 2、M 3 の順でメッセージを送信する必要があることを知る必要があり、コンポーネントに対する知識とメインフレームのトランザクション処理に関する知識との双方の知識が必要である。通常、クライアントプログラムの開発者はコンポーネントの知識はあるが、メインフレームの知識はない。

【 0 0 0 9 】

もう1つの問題は、メソッドMT 1 ~ MT 3 のパラメータに関してはその公開情報により型の情報は分かるが、個々のパラメータが、・入力パラメータ、・出力パラメータ、・入出力パラメータ、或いは・パラメータとして働かない、の何れの状態をとるか分からないため、クライアントプログラム 3 0 4 の開発者にとってクライアントプログラム 3 0 4 中のメソッドMT 1 ~ MT 3 の呼び出し前後に必要となるパラメータへの値の設定処理、パラメータからの値の取得処理の記述の作成が容易でないことである。例えば、メソッドMT 1 のパラメータP R 1 が入力パラメータであれば、メソッドMT 1 の呼び出し前にパラメータP R 1 へ値を設定する処理を記述する必要がある、出力パラメータであれば、メソッドMT 1 の呼び出し後にパラメータP R 1 の値を取得する処理を記述する必要がある、入出力パラメータであれば、メソッドMT 1 の呼び出し前にパラメータP R 1 へ値を設定する処理を記述し、呼び出し後にパラメータP R 1 の値を取得する処理を記述する必要がある。このようなパラメータの入出力種別を理解するには、メソッドMT 1 ~ MT 3 の起動順序を理解する場合と同様にコンポーネントに関する知識に加えて、メインフレーム 3 0 0 のトランザクション処理におけるパラメータの取り扱いに関する知識が必要である。

【 0 0 1 0 】

クライアントプログラム 3 0 4 の開発者がメインフレーム 3 0 0 の技術者に対して、メソッド MT 1 ～ MT 3 の起動順序をどうすべきか、各パラメータの入出力種別はどうなっているかを問い合わせることも考えられるが、既にソフトウェアコンポーネントの開発を終えているため、ソフトウェアコンポーネントの開発に関与したメインフレーム 3 0 0 の技術者がその時点で存在するとは限らず、問い合わせできない事態も考えられる。また、たとえ存在しても、問い合わせること自体が双方にとって面倒な作業となる。更に、他のメインフレームの技術者に問い合わせるとしても、ソフトウェアコンポーネントの開発に関与していない技術者では、どのパラメータが入力、出力になるかがわかっていても、またメッセージの送出順序がわかっていても、コンポーネントの理解がないため、クライアントプログラムの開発者に理解できるように説明することは難しい。

【 0 0 1 1 】

他方、特開平 1 1 - 2 2 4 1 8 4 号公報には、自動生成したオートメーションサーバが公開するデータベースのスキーマクラスとその属性、メソッドを利用する外部アプリケーションとなるクライアントプログラムを自動生成する技術が記載されているが、同公報に記載のクライアントプログラムはデータベースへのアクセスのインタフェースそのものであり、これ自体、もともと、オブジェクト指向データベースの段階で C ++ のクラスと 1 対 1 に対応する情報で、クライアント側に呼び出し順序（シーケンス）や、入力、出力、入出力の概念を要求しない。従って、同公報の技術では本発明が解決しようとする上記の課題を解決することはできない。

【 0 0 1 2 】

そこで本発明の目的は、メインフレーム上で動作するトランザクション処理プログラムにアクセスするソフトウェアコンポーネントの生成に必要な情報だけでなく、そのソフトウェアコンポーネントを利用するクライアントプログラムの生成に必要な情報も含む定義情報を入力して保存しておき、この定義情報に基づいてクライアントプログラムのソースコードを自動的に生成することによって、クライアントプログラムの開発者の負担を大幅に軽減することができるようにする

ことにある。

【 0 0 1 3 】

【課題を解決するための手段】

本発明のソフトウェアコンポーネント自動生成システムは、パラメータを含む一連のメッセージを受け付けて所定のトランザクションを実行するトランザクション処理プログラムが動作するメインフレームにアクセスするソフトウェアコンポーネント及び該ソフトウェアコンポーネントを利用するクライアントプログラムのソースコードを生成するシステムである。ここで、ソフトウェアコンポーネントとはCOM、Active Xなどにおけるコンポーネントのような外部から利用可能な形態を取るソフトウェア部品であり、ユーザが直接ロジックのコーディングを行えない形態を指し、クライアントプログラムとはこの部品を利用するユーザプログラムの一部として使われるプログラムのソースコードを指す。そして本発明は、前記ソフトウェアコンポーネントを生成するのに必要なモジュール情報、コンポーネント情報、メソッド情報およびパラメータ情報に加えて、前記クライアントプログラムを生成するのに必要となる情報を利用者入出力装置から入力してソフトウェアコンポーネント定義情報を生成するソフトウェアコンポーネント定義部と、前記ソフトウェアコンポーネント定義情報に基づいて前記ソフトウェアコンポーネントを生成するソフトウェアコンポーネント自動生成部と、前記ソフトウェアコンポーネント定義情報に基づいて前記クライアントプログラムのソースコードを生成するクライアントプログラム自動生成部とを備え、前記クライアントプログラムを生成するのに必要な情報として、前記トランザクション処理プログラムに与えるメッセージと1対1に対応する前記ソフトウェアコンポーネント中のメソッドの呼び出し順の情報を含み、前記クライアントプログラム自動生成部は、前記メソッドの呼び出し順の情報を参照して、クライアントプログラム中のメソッドの呼び出し部分の記述を生成するコンポーネント呼び出し生成手段を含んでいる。

【 0 0 1 4 】

また本発明は、前記クライアントプログラムを生成するのに必要な情報として、更に、前記トランザクション処理プログラムに与えるメッセージで指定するパ

ラメータの入出力種別の情報を含み、前記クライアントプログラム自動生成部は、前記パラメータの入出力種別の情報を参照して、入力となるパラメータを持つメソッドの呼び出し前に当該パラメータへ値を設定する処理を記述する前処理生成手段を含むように構成して良い。

【 0 0 1 5 】

また本発明は、前記クライアントプログラム自動生成部が、前記パラメータの入出力種別の情報を参照して、出力となるパラメータを持つメソッドの呼び出し後に当該パラメータの値を取得する処理を記述する後処理生成手段を含むように構成して良い。

【 0 0 1 6 】

また本発明は、前記クライアントプログラム自動生成部が、前記コンポーネント呼び出し生成手段、前記前処理生成手段、前記後処理生成手段に加えて更に、クライアントモジュールの骨格を生成するクライアントモジュール生成手段と、入力、出力、入出力として使われるパラメータとして使うための変数の宣言を生成するパラメータ定義生成手段を備えるようにして良い。

【 0 0 1 7 】

また本発明は、前記クライアントプログラム自動生成部が、クライアントプログラム生成ルールに従ってクライアントプログラムの生成を行うものであって良い。この場合、クライアントプログラム生成ルールは、前記パラメータ定義生成手段によって参照されるパラメータ定義生成ルール、前記前処理生成手段によって参照される前処理生成ルール、前記後処理生成手段によって参照される後処理生成ルール、前記コンポーネント呼び出し生成手段によって参照されるコンポーネント呼び出し生成ルール、および前記クライアントモジュール生成手段によって参照されるクライアントモジュール生成ルールから構成されていて良い。

【 0 0 1 8 】

【作用】

本発明のソフトウェアコンポーネント自動生成システムにあっては、ソフトウェアコンポーネント定義部が、ソフトウェアコンポーネントを生成するのに必要な情報に加えて、そのソフトウェアコンポーネントを利用するクライアントプロ

グラムを生成するのに必要となる情報も利用者入出力装置から入力してソフトウェアコンポーネント定義情報を生成し、ソフトウェアコンポーネント自動生成部がソフトウェアコンポーネント定義情報に基づいてソフトウェアコンポーネントを生成する一方、クライアントプログラム自動生成部がソフトウェアコンポーネント定義情報に基づいてクライアントプログラムのソースコードを生成する。この際、クライアントプログラム自動生成部内のコンポーネント呼び出し生成手段が、前記ソフトウェアコンポーネント定義情報からトランザクション処理プログラムに与えるメッセージと 1 対 1 に対応する前記ソフトウェアコンポーネント中のメソッドの呼び出し順の情報を参照して、クライアントプログラム中のメソッドの呼び出し部分の記述を生成する。これによって、クライアントプログラムの開発者はメソッドの呼び出し部分を自ら記述する必要がなく、メインフレームの技術者にメソッドの呼び出し順を問い合わせる必要すらなくなる。

【 0 0 1 9 】

また、クライアントプログラムを生成するのに必要な情報として、トランザクション処理プログラムに与えるメッセージで指定するパラメータの入出力種別の情報が含まれる場合、クライアントプログラム自動生成部内の前処理生成手段が、パラメータの入出力種別の情報を参照して、入力となるパラメータを持つメソッドの呼び出し前に当該パラメータへ値を設定する処理を記述し、後処理生成手段が、パラメータの入出力種別の情報を参照して、出力となるパラメータを持つメソッドの呼び出し後に当該パラメータの値を取得する処理を記述する。前処理生成手段が記述するパラメータへ値を設定する処理の記述は、クライアントプログラムのソースコードの生成時点では値の入力元が不明なため、そのことを示す所定の記号（例えば「？」）を使って、例えば、「パラメータ名＝？」といった記述として生成する。同様に後処理生成手段が記述するパラメータの値を取得する処理の記述は、その時点では値の出力先が不明なため、そのことを示す所定の記号（例えば「？」）を使って、例えば、「？＝パラメータ名」といった記述として生成する。これによって、クライアントプログラム開発者は、どういう情報を入力して、どういう情報が出力されるかを容易に理解することができる。なお、？の箇所には、その後、実際の入力元、出力先となる箇所の情報（例えばクラ

クライアントプログラム開発者が設計したGUI画面の入力欄、出力欄の情報などが手作業で設定される。

【0020】

【発明の実施の形態】

次に本発明の実施の形態の例について図面を参照して詳細に説明する。

【0021】

図1を参照すると、本実施の形態にかかるソフトウェアコンポーネント自動生成システムは、利用者入出力装置200を通じて利用者から入力される情報に基づいてソフトウェアコンポーネント定義情報50を生成するソフトウェアコンポーネント定義部20を有する。このソフトウェアコンポーネント定義部20は、ソフトウェアコンポーネントだけでなくクライアントプログラムも生成できるようにする為に、定義情報のコンポーネントを生成するのに必要な情報だけでなくクライアントプログラムを生成するのに必要な情報51も同時に定義することができる。また、既に定義された情報にクライアントプログラムを生成するのに必要な情報51を追加することができる。

【0022】

クライアントプログラムを生成するのに必要な定義情報51の例としては、メソッドの呼び出し順の情報がある。また、他の例として、メソッドの各パラメータが、・入力パラメータ、・出力パラメータ、・入出力パラメータ、或いは・パラメータとして働かない、の何れの状態をとるかを定義する情報がある。

【0023】

本実施の形態にかかるソフトウェアコンポーネント自動生成システムは、プログラム生成手段として、ソフトウェアコンポーネント自動生成部30とクライアントプログラム自動生成部40との2つの生成手段を備えている。ソフトウェアコンポーネント自動生成部30は、ソフトウェアコンポーネント定義情報50の定義情報に基づいてソフトウェアコンポーネントモジュール70を自動的に生成する手段である。他方、クライアントプログラム自動生成部40は、ソフトウェアコンポーネント定義情報50中に含まれるクライアントプログラムを生成するのに必要な情報51を活用することにより、ソフトウェアコンポーネントモジュ

ール 7 0 を利用するクライアントプログラムのソースコード 9 0 を自動的に生成する手段である。

【 0 0 2 4 】

クライアントプログラム自動生成部 4 0 は、定義情報抽出手段 4 1、クライアントモジュール生成手段 4 2、パラメータ定義生成手段 4 3、前処理生成手段 4 4、コンポーネント呼び出し生成手段 4 5 および後処理生成手段 4 6 から構成される。定義情報抽出手段 4 1 によってソフトウェアコンポーネント定義情報 5 0 から、生成対象とするクライアントプログラムの生成に必要な定義情報を抽出し、この抽出された定義情報を基に、クライアントモジュール生成手段 4 2 がモジュール全体の枠組みを生成する。そして、その枠組み内に、パラメータ定義生成手段 4 3 が入力、出力、入出力として使われるパラメータとして使うための変数の宣言やパラメータの初期化処理などを生成し、前処理生成手段 4 4 がコンポーネントの宣言やコンポーネントの呼び出し前に必要な入力パラメータやプロパティへの値の設定処理などを生成し、コンポーネント呼び出し生成手段 4 5 がメソッドなどの呼び出し処理などを生成し、後処理生成手段 4 6 が出力パラメータの値の取得処理などコンポーネントの呼び出し後に必要な処理を生成することにより、クライアントプログラムのソースコード 9 0 を生成する。

【 0 0 2 5 】

図 1 のクライアントプログラムのソースコード 9 0 中には、特徴的な記述が例示されている。符号 9 1 の部分は、ソフトウェアコンポーネント定義情報 5 0 中のクライアントプログラムの生成に必要な定義情報 5 1 に基づいてコンポーネント呼び出し生成手段 4 5 が自動生成したメソッドの呼び出し部分の記述である。この例は、接続、送信、受信、切断の手順で呼び出さなければならないという定義情報に基づき、接続メソッド、送信メソッド、受信メソッド、切断メソッドの順でメソッドを呼び出すように記述されている。

【 0 0 2 6 】

また、符号 9 2 の部分は、メソッドのパラメータの種別情報に基づいて前処理生成手段 4 4 が自動生成したパラメータへの値の設定処理の記述である。「P a r a m 1 = ?」、「P a r a m 2 = ?」（P a r a m 1、P a r a m 2 はパラメ

ータ名) という記述から、クライアントプログラムの開発者は、当該パラメータは入力パラメータで、メソッドの呼び出し前に値を設定する必要があることを容易に認識することができる。ここで、? の箇所には後に手作業で具体的な入力元の情報 (例えば G U I における所定の入力欄の情報など) が設定される。この際、パラメータ名が適切な意味のある名称で定義されていれば、コンポーネントの使用の確認を最低限にして作業が可能となる。

【 0 0 2 7 】

また、符号 9 3 の部分は、メソッドのパラメータの種別情報に基づいて後処理生成手段 4 6 が自動生成したパラメータの値の取得処理の記述である。「? = P a r a m 3」、「? = P a r a m 4」という記述から、クライアントプログラムの開発者は、当該パラメータは出力パラメータで、メソッドの呼び出し後に値を取得する必要があることを容易に認識することができる。ここで、? の箇所には後に手作業で具体的な出力先の情報 (例えば G U I における所定の出力欄の情報など) が設定される。この際、パラメータ名が適切な意味のある名称で定義されていれば、コンポーネントの使用の確認を最低限にして作業が可能となる。

【 0 0 2 8 】

クライアントプログラム自動生成部 4 0 で自動再生されるクライアントプログラムのソースコード 9 0 は、ソフトウェアコンポーネント自動生成部 3 0 で生成されるソフトウェアコンポーネントモジュール 7 0 を利用するクライアントプログラムをクライアントプログラム開発者が開発する際のひな型 (テンプレート) となるものである。開発者は、自動生成されたクライアントプログラムのソースコード 9 0 中の未定義の部分 (例えば符号 9 2、9 3 中の ? の箇所など) を完成させ、また、必要に応じて V i s u a l B a s i c などを使ってクライアントプログラムの G U I 部分などを作成して、クライアントプログラムのソースコード 9 0 を完成する。

【 0 0 2 9 】

【実施例】

次に本発明の実施例について図面を参照して詳細に説明する。

【 0 0 3 0 】

図 2 を参照すると、本発明の実施例にかかるソフトウェアコンポーネント自動生成システムは、処理装置 1 0 と記憶装置 1 0 0 と利用者入出力装置 2 0 0 とを主要部として備えている。利用者入出力装置 2 0 0 は、例えばキーボードやマウス等の入力装置と、LCD等の表示装置とで構成される。記憶装置 1 0 0 は、半導体メモリや磁気ディスク装置等で構成されており、ソフトウェアコンポーネント生成ルール 6 0 及びクライアントプログラム生成ルール 8 0 を事前に記憶すると共に、システムの処理の過程でソフトウェアコンポーネント定義情報 5 0 を記憶し、また生成したソフトウェアコンポーネントモジュール 7 0 及びクライアントプログラムソースコード 9 0 を記憶する。処理装置 1 0 はCPU及びその上で動作するソフトウェアによって構成され、それらによってソフトウェアコンポーネント定義部 2 0、ソフトウェアコンポーネント自動生成部 3 0 及びクライアントプログラム自動生成部 4 0 の 3 つの機能部を提供している。

【 0 0 3 1 】

ソフトウェアコンポーネント定義部 2 0 は一種のエディタであり、ソフトウェアコンポーネント及びそれを利用するクライアントプログラムを生成するのに必要な各種の情報を利用者が利用者入出力装置 2 0 0 を通じて定義するためのユーザインタフェースを提供する。ソフトウェアコンポーネント定義部 2 0 は、画面からの入力またはそれに代わる情報のインポートを行うことによってソフトウェアコンポーネント定義情報 5 0 を得る。得たソフトウェアコンポーネント定義情報 5 0 は記憶装置 1 0 0 に格納され、ソフトウェアコンポーネント自動生成部 3 0 及びクライアントプログラム自動生成部 4 0 によって適宜に参照される。

【 0 0 3 2 】

クライアントプログラム生成ルール 8 0 は、ソフトウェアコンポーネント定義情報 5 0 に基づいてクライアントプログラムを生成する際のルールを記述している。具体的には、先ず、生成対象となる定義情報の種類やその生成の仕方などを記述している。具体的な定義情報の値はソフトウェアコンポーネント定義情報 5 0 から取得する。例えば、或るルールではクライアントプログラムの骨格を定義することが記されており、その際に必要なコンポーネント名はソフトウェアコンポーネント定義情報 5 0 から取得する。同様に、他のルールでは、パラメータ定

義、前処理、コンポーネント呼び出し処理、後処理などを定義すべきこと及びその定義方法が指定され、それらの生成に必要なパラメータ名や型などの具体値はソフトウェアコンポーネント定義情報 5 0 から取得する。

【 0 0 3 3 】

クライアントプログラム生成ルール 8 0 では、また、定義情報に関連して出力される情報とその出力位置または出力順序を記述している。例えば、前処理で生成されたパラメータへの値の設定処理は、そのパラメータを含むメソッドの呼び出し生成の直前に出力すべきことや、後処理で生成されたパラメータからの値の取得処理は、そのパラメータを含むメソッドの呼び出し直後に出力すべきこと等が記述されている。

【 0 0 3 4 】

なお、クライアントプログラム生成ルール 8 0 には、定義情報に関連せずに出力される情報とその出力位置や出力順序の情報も記述されている。例えば、エラー処理や固定で行われる前処理、後処理などである。

【 0 0 3 5 】

ソフトウェアコンポーネント生成ルール 6 0 は、ソフトウェアコンポーネント定義情報 5 0 に基づいてソフトウェアコンポーネントを生成する際のルールを記述している。クライアントプログラム生成ルール 8 0 と同様に、生成対象となる定義情報の種類やその定義の仕方、定義情報に関連して出力される情報とその出力位置または出力順序、定義情報に関連せずに出力される情報とその出力位置または出力順序などが記述されている。

【 0 0 3 6 】

ソフトウェアコンポーネント自動生成部 3 0 は、ソフトウェアコンポーネント定義情報 5 0 とソフトウェアコンポーネント生成ルール 6 0 とに基づいて、ソフトウェアコンポーネントモジュール 7 0 を生成する。生成されたソフトウェアコンポーネントモジュール 7 0 は、本実施例の場合、記憶装置 1 0 0 に格納される。

【 0 0 3 7 】

クライアントプログラム自動生成部 4 0 は、ソフトウェアコンポーネント定義

情報 5 0 とクライアントプログラム生成ルール 8 0 とに基づいて、ソフトウェアコンポーネント自動生成部 3 0 が生成したソフトウェアコンポーネントを利用するクライアントプログラムのソースコード 9 0 を生成する。生成されたクライアントプログラムのソースコード 9 0 は、本実施例の場合、記憶装置 1 0 0 に格納される。

【 0 0 3 8 】

以下、本実施例のソフトウェアコンポーネント自動生成システムの各部を詳細に説明する。

【 0 0 3 9 】

(1) ソフトウェアコンポーネント定義部 2 0

一般に 1 つのソフトウェアコンポーネントモジュール中には複数のソフトウェアコンポーネントが存在し得る。また、1 つのソフトウェアコンポーネント中には複数のメソッドが存在し得る。更に、1 つのメソッド中には複数のパラメータが存在し得る。このため、1 つのソフトウェアコンポーネントモジュールの生成に必要な定義情報は、一般に図 3 に示されるような階層構造となる。つまり、1 つのモジュール情報 5 1 の配下に 1 以上のコンポーネント情報 5 2 があり、各コンポーネント情報 5 2 の配下に 1 以上のメソッド情報 5 3 があり、各メソッド情報 5 3 の配下に 1 以上のパラメータ情報 5 4 がある。

【 0 0 4 0 】

ソフトウェアコンポーネント定義部 2 0 は、以上のような階層構造に合わせて、図 4 に示されるように、コンポーネント情報定義手段 2 1、メソッド情報定義手段 2 2 及びパラメータ情報定義手段 2 3 の 3 つの定義手段を有する。コンポーネント情報定義手段 2 1 は、生成対象となるソフトウェアコンポーネントモジュールに含まれる個々のソフトウェアコンポーネント毎に、コンポーネント固有の情報（モジュール情報 5 1 及びコンポーネント情報 5 2）を定義する際に利用され、メソッド情報定義手段 2 2 は、コンポーネント情報 5 2 の配下にメソッド固有の情報（メソッド情報 5 3）を定義する際に利用され、パラメータ情報定義手段 2 3 は、メソッド情報 5 3 の配下にパラメータ固有の情報（パラメータ情報 5 4）を定義する際に利用される。

【 0 0 4 1 】

ここで、本実施例においては、各定義手段 2 1 ~ 2 3 は、ソフトウェアコンポーネントモジュールだけでなくクライアントプログラムも生成できるように、クライアントプログラムの生成に必要な情報を定義するためにも利用される。つまり、コンポーネント情報定義手段 2 1 は、クライアントプログラムの生成に必要な情報を含んだコンポーネント固有の情報の定義を行い、メソッド情報定義手段 2 2 は、クライアントプログラムの生成に必要な情報を含んだメソッド固有の情報の定義を行い、パラメータ情報定義手段 2 3 は、クライアントプログラムの生成に必要な情報を含んだパラメータ固有の情報の定義を行う。従って、図 2 に示されるような階層構造をとるソフトウェアコンポーネント定義情報には、ソフトウェアコンポーネントだけでなく、それを利用するクライアントプログラムを生成するのに必要な情報も含まれている。

【 0 0 4 2 】

モジュール情報 5 1、コンポーネント情報 5 2、メソッド情報 5 3 及びパラメータ情報 5 4 に含まれる情報の一部の例を図 5 に示す。

【 0 0 4 3 】

図 5 を参照すると、モジュール情報 5 1 には、ソフトウェアコンポーネントモジュールの名前であるモジュール名 5 1 1 が含まれる。

【 0 0 4 4 】

コンポーネント情報 5 2 には、ソフトウェアコンポーネントの名前であるコンポーネント名 5 2 1、コンポーネントが持つ初期値（デフォルトパラメータ）を示すコンポーネント属性情報 5 2 2 が含まれる。このコンポーネント属性情報 5 2 2 の一例としては、メインフレームへの接続情報がある。メインフレームへの接続情報とは例えば以下の情報のことである。

- ・メインフレームの I P アドレス = 111.111.111.111
- ・メインフレームのポート番号 = 8000
- ・送信時のタイムアウト値 = 3
- ・受信時のタイムアウト値 = 3

【 0 0 4 5 】

メソッド情報 5 3 には、メソッドの名前であるメソッド名 5 3 1、メソッドタイプ、メソッドのリターン値の型などを示すメソッド属性情報 5 3 2、送信型、受信型、送受信型の種別を示すメソッドの呼び出し形式の情報 5 3 3、複数のメソッドがある場合に、どのような順序で呼び出す必要があるかを示すメソッドの呼び出し順の情報 5 3 4 が含まれる。なお、メソッドの呼び出し順の情報 5 3 4 を定義する方法としては、各メソッド情報 5 3 中に自メソッドの呼び出し順番を明示する方法でも良く、コンポーネント情報 5 2 配下の複数のメソッド情報 5 3 の記述順序をメソッドの呼び出し順番と一致させておく方法を採用しても良い。

【 0 0 4 6 】

パラメータ情報 5 4 には、パラメータの名前であるパラメータ名 5 4 1、パラメータの型 5 4 2、パラメータの入出力種別の情報 5 4 3 が含まれる。パラメータ名は適切な意味のある名称で定義しておくことが望ましい。パラメータの入出力種別の情報 5 4 3 では、当該パラメータが、・入力パラメータ、・出力パラメータ、・入出力パラメータ、・パラメータとして機能しない、の 4 つの状態の内の何れの状態をとるかが定義される。

【 0 0 4 7 】

(2) ソフトウェアコンポーネント自動生成部 3 0 及びソフトウェアコンポーネント生成ルール 6 0

ソフトウェアコンポーネント自動生成部 3 0 は、図 6 に示されるように、コンポーネント生成手段 3 1、メソッド生成手段 3 2、パラメータ生成手段 3 3 及びモジュール生成手段 3 4 から構成される。また、ソフトウェアコンポーネント生成ルール 6 0 は、ソフトウェアコンポーネント自動生成部 3 0 がソフトウェアコンポーネント定義情報 5 0 に基づいてソフトウェアコンポーネントモジュール 7 0 を生成する際に使用する各種のルールの集合であり、図 7 に示されるように、コンポーネント生成ルール 6 1、メソッド生成ルール 6 2、パラメータ生成ルール 6 3 およびモジュール生成ルール 6 4 を含む。更に、コンポーネント生成ルール 6 1 は、プロパティ生成ルール 6 1 1、初期値生成ルール 6 1 2 及びコンポーネント骨格生成ルール 6 1 3 を含み、メソッド生成ルール 6 2 は、メソッド骨格生成ルール 6 2 1 及びメソッドロジック生成ルール 6 2 2 を含む。

【 0 0 4 8 】

パラメータ生成手段 3 3 は、メソッド生成手段 3 2 からのパラメータ生成要求に応じ、ソフトウェアコンポーネント定義情報 5 0 及びパラメータ生成ルール 6 3 を適宜参照して、パラメータ部分を生成する手段である。

【 0 0 4 9 】

メソッド生成手段 3 2 は、コンポーネント生成手段 3 1 からのメソッド生成要求に応じ、ソフトウェアコンポーネント定義情報 5 0 及びメソッド生成ルール 6 2 を適宜参照して、メソッド部分を生成する手段である。

【 0 0 5 0 】

コンポーネント生成手段 3 1 は、モジュール生成手段 3 4 からのコンポーネント生成要求に応じ、ソフトウェアコンポーネント定義情報 5 0 及びコンポーネント生成ルール 6 1 を適宜参照して、コンポーネント部分を生成する手段である。

【 0 0 5 1 】

モジュール生成手段 3 4 は、ソフトウェアコンポーネント定義情報 5 0 及びモジュール生成ルール 6 4 を適宜参照し、またコンポーネント生成手段 3 1 を適宜呼び出すことにより、ソフトウェアコンポーネントモジュール 7 0 全体を生成する手段である。

【 0 0 5 2 】

以下、モジュール生成手段 3 4、コンポーネント生成手段 3 1、メソッド生成手段 3 2、パラメータ生成手段 3 3 の順に、その機能をルールの内容例と共に詳細に説明する。

【 0 0 5 3 】

(2 - 1) モジュール生成手段 3 4

図 8 にモジュール生成手段 3 4 の処理例を示す。モジュール生成手段 3 4 は、先ずソフトウェアコンポーネント定義情報 5 0 から、モジュール情報 5 1 中のモジュール名 5 1 1 と、モジュール情報 5 1 配下の全てのコンポーネント情報 5 2 中のコンポーネント名 5 2 1 とを取得する (S 1 1) 。次に、取得したモジュール名 5 1 1 及びコンポーネント名 5 2 1 と、モジュール生成ルール 6 4 中に存在するモジュールのプロジェクトファイルの生成ルールとに基づいて、モジュール

のプロジェクトファイルを生成する（S 1 2）。

【 0 0 5 4 】

モジュールのプロジェクトファイルの生成ルールには、例えば以下のようなルールが予め記述されている。

「テンプレート「Class =コンポーネント名；コンポーネント名.cls」中のコンポーネント名の箇所にプロジェクトを構成するコンポーネント名を設定し、テンプレート「Name=" モジュール名"、ExeName32 =" モジュール名.dll"」中のモジュール名の箇所にプロジェクトのモジュール名を設定し、プロジェクトファイルを生成する」

【 0 0 5 5 】

従って、ステップ S 1 1 で取得したモジュール名 5 1 1 をComponentModuleName、コンポーネント名 5 2 1 をComponent1、Component2、Component3の3つとすると、以下のようなプロジェクトファイルが生成される。

```
Class =Component1; Component1.cls
Class =Component1; Component1.cls
Class =Component1; Component1.cls
Name=" ComponentModuleName "
ExeName32 =" ComponentModuleName.dll "
```

ここで、Component1、Component2、Component3はコンポーネントのファイル、ComponentModuleName.dll はソフトウェアコンポーネントモジュールのファイルである。

【 0 0 5 6 】

次にモジュール生成手段 3 4 は、コンポーネント生成手段 3 1 に対して、モジュール内のコンポーネントの生成処理をコンポーネントの数だけ呼び出す（S 1 3）。各呼び出し時には、コンポーネント名 5 2 1 が渡される。呼び出しによって生成されたコンポーネントはコンポーネントのファイルに格納される。

【 0 0 5 7 】

（2 - 2）コンポーネント生成手段 3 1

図 9 にコンポーネント生成手段 3 1 の処理例を示す。コンポーネント生成手段

3 1 は、モジュール生成手段 3 4 からコンポーネント名 5 2 1 を取得すると（S 2 1）、先ず、コンポーネント骨格生成ルール 6 1 3 に基づき、コンポーネントの骨格を生成する（S 2 2）。この処理は取得したコンポーネント名 5 2 1 に対応する定型的なヘッダを生成する処理に相当する。

【 0 0 5 8 】

コンポーネント骨格生成ルール 6 1 3 には、例えば以下のようなルールが記述されているので、このルールに従ってコンポーネントの骨格を生成する。

「以下のテンプレートのコンポーネント名の箇所に取得したコンポーネント名を設定する。

```
VERSION 1.0 CLASS
```

```
BEGIN
```

```
MultiUse=-1 ' True
```

```
Persistable =0 ' NotPersistable
```

```
DataBindingBehavior =0 ' vbNone
```

```
DataSourceBehavior=0 ' vbNone
```

```
MTSTransactionMode =0 ' NotAnMTSObject
```

```
END
```

```
Attribute VB-Name =" コンポーネント名"
```

```
Attribute VB-GlobalNameSpace=False
```

```
Attribute VB-ClobalNameSpace=False
```

```
Attribute VB-Creatable=True
```

```
Attribute VB-Exposed=True
```

【 0 0 5 9 】

次にコンポーネント生成手段 3 1 は、ソフトウェアコンポーネント定義情報 5 0 から当該コンポーネントのコンポーネント属性情報 5 2 を取得し（S 2 3）、プロパティ生成ルール 6 1 1 に基づいて、取得したコンポーネント属性情報の数だけコンポーネントのプロパティを生成する（S 2 4）。また、初期値生成ルール 6 1 2 に基づいて、取得したコンポーネント属性情報の数だけコンポーネントの初期値を生成する（S 2 5）。

【 0 0 6 0 】

プロパティ生成ルール 6 1 1 としては、IP アドレス用、ポート番号用、タイムアウト用など、ソフトウェアコンポーネント定義情報 5 0 中に存在し得るコンポーネント属性情報の各種類に応じたテンプレートが用意されている。各テンプレートは例えば、

```
Public AAA As String
```

のような形式である。コンポーネント生成手段 3 1 は、取得したコンポーネント属性情報の名前 (AAA) に対応するテンプレートをプロパティ生成ルール 6 1 1 から取得することで、各コンポーネント属性情報と型に対応するプロパティを生成する。これによって、接続先サーバ名、メインフレームの IP アドレス、メインフレームのポート番号、送信時のタイムアウト値、受信時のタイムアウト値といったコンポーネント属性情報が取得された場合、それぞれに対応するプロパティが生成される。

【 0 0 6 1 】

また、初期値生成ルール 6 1 2 としては、例えば以下のようなルールが用意されている。

「以下のテンプレートの AAA = BBB の箇所に、各コンポーネント属性情報に対応するプロパティ名とその値を、プロパティ名 = 値の形式で、初期値として設定する。

```
Private Sub Class-Initialize()
```

```
AAA = BBB
```

```
End Sub 」
```

【 0 0 6 2 】

このような初期値生成ルール 6 1 2 と取得されたコンポーネント属性情報とから、例えば以下のような初期値生成処理が生成される。

```
Private Sub Class-Initialize()
```

```
MainFrameIP = " 111.111.111.111 "
```

```
MainFramePort = 8000
```

```
RcvTimeOut = 3
```

SndTimeOut=3

End Sub

【 0 0 6 3 】

次にコンポーネント生成手段 3 1 は、当該コンポーネントに存在する全てのメソッド情報 5 3 中のメソッド名 5 3 1 をソフトウェアコンポーネント定義情報 5 0 から取り出し、当該コンポーネント中に定義されたメソッドの数だけメソッド生成手段 3 2 を呼び出す (S 2 6)。各呼び出し時には、メソッド名とそのメソッドを有するコンポーネント名とが渡される。

【 0 0 6 4 】

(2-3) メソッド生成手段 3 2

図 1 0 にメソッド生成手段 3 2 の処理例を示す。メソッド生成手段 3 2 は、コンポーネント生成手段 3 1 からメソッド名とそのメソッドを有するコンポーネント名を取得すると (S 3 1)、先ず、ソフトウェアコンポーネント定義情報 5 0 とメソッド骨格生成ルール 6 2 1 とに基づき、メソッドの骨格を生成する (S 3 2)。

【 0 0 6 5 】

メソッド骨格生成ルール 6 2 1 には、例えば以下のようなルールが用意されている。

「メソッド名にメソッドパラメータとその型とリターン値の型を以下のフォーマットで連結して出力する。

```
Public Function メソッド名 (全メソッドパラメータ) As リターン値の型
メソッドの処理
End Function
```

なお、メソッドパラメータの生成ではパラメータ生成手段 3 3 を呼び出す。呼び出し時には、パラメータ名と、そのパラメータを有するメソッド名と、そのメソッドを有するコンポーネント名とを指定する。

【 0 0 6 6 】

以上の処理により、例えば、メソッド名を Method1、リターン値の型を Long、パラメータ生成手段 3 3 を呼び出して得たメソッドパラメータを Param1 As Inte

ger、Param2 As Stringの2つとすると、以下のようなメソッドの骨格が生成される。

```
Public Function Method1(Param1 As Integer,Param2 As String) As Long
```

```
メソッドの処理
```

```
End Function
```

【 0 0 6 7 】

次にメソッド生成手段32は、ソフトウェアコンポーネント定義情報50とメソッドロジック生成ルール622とに基づき、メソッドの処理を生成する(S33)。メインフレームへアクセスする機能を持つメソッドの場合、生成するメソッドの処理は、メインフレームへの通信処理となる。メインロジック生成ルール622には、メインフレームの外部インタフェースに適合した通信処理を記述したテンプレートが各メソッド毎、各メソッドの呼び出し形式毎に用意されているので、当該メソッドの呼び出し形式に対応するテンプレートを選択し、そのテンプレートにパラメータ名を設定することで、メソッドの処理を生成する。生成はパラメータに依存した情報はパラメータの数だけ生成を繰り返し、メソッドに固有の情報は1回だけ生成する。送信型、受信型、送受信型の例を以下に示す。

○送信型

```
Public Sub Method1 (...パラメータ...)
```

```
WbRet =Send (...パラメータ...)
```

```
End
```

○受信型

```
Public Sub Method2 (...パラメータ...)
```

```
WbRet =Recv (...パラメータ...)
```

```
End
```

○送受信型

```
Public Sub Method3 (...パラメータ...)
```

```
WbRet =Send (...パラメータ...)
```

WbRet =Recv (…パラメータ…)

End

【 0 0 6 8 】

(2 - 4) パラメータ生成手段 3 3

図 1 1 にパラメータ生成手段 3 3 の処理例を示す。パラメータ生成手段 3 3 は、メソッド生成手段 3 2 からパラメータ名とそれが属するメソッド名及びコンポーネント名を取得すると (S 4 1)、ソフトウェアコンポーネント定義情報 5 0 中から当該コンポーネント情報かつメソッド情報配下の該当パラメータの型 5 4 2 を取得し、パラメータ生成ルール 6 3 に従ってメソッドパラメータを生成する (S 4 2)。

【 0 0 6 9 】

パラメータ生成ルール 6 3 には、パラメータ名とその型を As で連結してメソッドパラメータを生成する旨のルールが用意されている。従って、パラメータ名を Param1、その型を String とすると、以下のようなメソッドパラメータが呼び出し元のメソッド生成手段 3 2 に返される。

Param1 As String

【 0 0 7 0 】

(3) クライアントプログラム自動生成部 4 0 及びクライアントプログラム生成ルール 8 0

クライアントプログラム自動生成部 4 0 は、図 1 2 に示されるように、定義情報抽出手段 4 1、クライアントモジュール生成手段 4 2、パラメータ定義生成手段 4 3、前処理生成手段 4 4、コンポーネント呼び出し生成手段 4 5 及び後処理生成手段 4 6 から構成される。また、クライアントプログラム生成ルール 8 0 は、クライアントプログラム自動生成部 4 0 がソフトウェアコンポーネント定義情報 5 0 に基づいてクライアントプログラムのソースコード 9 0 を生成する際に使用する各種のルールの集合であり、図 1 3 に示されるように、クライアントモジュール生成ルール 8 1、パラメータ定義生成ルール 8 2、前処理生成ルール 8 3、コンポーネント呼び出し生成ルール 8 4 及び後処理生成ルール 8 5 を含む。更に、クライアントモジュール生成ルール 8 1 は、クライアントプログラム骨格生

成ルール 8 1 1 及びモジュールのプロジェクトファイルの生成ルール 8 1 2 を含み、前処理生成ルール 8 3 は、コンポーネント宣言生成ルール 8 3 1 及び入力パラメータ設定処理生成ルール 8 3 2 を含む。

【0071】

定義情報抽出手段 4 1 は、各手段 4 2 ～ 4 6 で必要となる定義情報をソフトウェアコンポーネント定義情報 5 0 から抽出する手段である。

【0072】

クライアントモジュール生成手段 4 2 は、ソフトウェアコンポーネント定義情報 5 0 及びクライアントモジュール生成ルール 8 1 を適宜参照し、またパラメータ定義生成手段 4 3、前処理生成手段 4 4、コンポーネント呼び出し生成手段 4 5 及び後処理生成手段 4 6 を適宜呼び出すことにより、クライアントプログラムのソースコード 9 0 全体の枠組みを生成する手段である。

【0073】

パラメータ定義生成手段 4 3 は、クライアントモジュール生成手段 4 2 からのパラメータ定義生成要求に応じ、ソフトウェアコンポーネント定義情報 5 0 及びパラメータ定義生成ルール 8 2 を適宜参照して、クライアントプログラムから呼び出すメソッドに対応するパラメータの宣言および定義を生成する手段である。

【0074】

前処理生成手段 4 4 は、クライアントモジュール生成手段 4 2 からの前処理生成要求に応じ、ソフトウェアコンポーネント定義情報 5 0 及び前処理生成ルール 8 3 を適宜参照して、メソッドを呼び出す前に行わなければならないパラメータやプロパティへの値の設定などのソフトウェアコンポーネントを呼び出す前に行う必要のある前処理を生成する手段である。固定で必要となるその他の初期化処理もここで生成する。

【0075】

コンポーネント呼び出し生成手段 4 5 は、クライアントモジュール生成手段 4 2 からのコンポーネント呼び出し生成要求に応じ、ソフトウェアコンポーネント定義情報 5 0 及びコンポーネント呼び出し生成ルール 8 4 を適宜参照して、ソフトウェアコンポーネントのメソッドなどを呼び出す部分を生成する手段である。

【 0 0 7 6 】

後処理生成手段 4 6 は、クライアントモジュール生成手段 4 2 からの後処理生成要求に応じ、ソフトウェアコンポーネント定義情報 5 0 及び後処理生成ルール 8 5 を適宜参照して、メソッドを呼び出した後に行わなければならないパラメータの値の取得などのソフトウェアコンポーネントを呼び出した後に行う必要のある後処理を生成する手段である。エラー処理や固定で必要となるその他の後処理もここで生成する。

【 0 0 7 7 】

以下、クライアントモジュール生成手段 4 2、パラメータ定義生成手段 4 3、前処理生成手段 4 4、コンポーネント呼び出し生成手段 4 5、後処理生成手段 4 6 の順に、その機能をルールの内容例と共に詳細に説明する。

【 0 0 7 8 】

(3 - 1) クライアントモジュール生成手段 4 2 及びクライアントモジュール生成ルール 8 1

図 1 4 にクライアントモジュール生成手段 4 2 の処理例を示す。クライアントモジュール生成手段 4 2 は、先ずソフトウェアコンポーネント定義情報 5 0 から定義情報抽出手段 4 1 を通じてモジュール情報 5 1 配下の全てのコンポーネント情報 5 2 中のコンポーネント名 5 2 1 を取得し、この取得したコンポーネント名 5 2 1 と、クライアントプログラム骨格生成ルール 8 1 1 とに基づいて、各コンポーネントに対応するクライアントプログラムの骨格を生成する (S 5 1) 。

【 0 0 7 9 】

クライアントプログラム骨格生成ルール 8 1 1 には、例えば以下のようなルールが予め記述されている。

「各コンポーネント毎に、以下のテンプレートのコンポーネント名の箇所に取得したコンポーネント名を設定し、クライアントプログラムの骨格を生成する。

```
Dim コンポーネント名 As Object
```

```
Public Sub Main()
```

```
End Sub 」
```

【 0 0 8 0 】

従って、取得したコンポーネント名がComponent1、Component2、Component3の3つとすると、以下の3つのクライアントプログラムの骨格が生成される。

Component1に対応するクライアントプログラムの骨格

```
Dim Component1 As Object
```

```
Public Sub Main()
```

```
End Sub
```

Component2に対応するクライアントプログラムの骨格

```
Dim Component2 As Object
```

```
Public Sub Main()
```

```
End Sub
```

Component3に対応するクライアントプログラムの骨格

```
Dim Component3 As Object
```

```
Public Sub Main()
```

```
End Sub
```

【 0 0 8 1 】

次に、クライアントモジュール生成手段42は、それぞれのコンポーネント毎に、定義情報抽出手段41を通じてソフトウェアコンポーネント定義情報50を適宜参照して、パラメータ定義手段43、前処理生成手段44、コンポーネント呼び出し生成手段45、後処理生成手段46を呼び出し、それぞれのコンポーネント毎に生成された骨格の中の適所（つまりルールにおいて指定された各情報の出力位置または出力順序に適合する箇所）に、パラメータ定義、前処理、コンポーネント呼び出し、後処理の各プログラムを生成することで、コンポーネントを呼び出すためのプログラムの生成を行う（S52）。

【 0 0 8 2 】

次に、クライアントモジュール生成手段42は、ソフトウェアコンポーネント

定義情報 5 0 とモジュールのプロジェクトファイルの生成ルール 8 1 2 とに基づいて、プロジェクトを構成するコンポーネント名等を記述したプロジェクトファイル(userapp.vbp)を生成する(S 5 3)。

【 0 0 8 3 】

モジュールのプロジェクトファイルの生成ルール 8 1 2 には、プロジェクトファイルのテンプレートが用意されているので、クライアントモジュール生成手段 4 2 は、そのテンプレート中にコンポーネント名を設定することで、プロジェクトファイルを生成することができる。例えば、コンポーネント名をComponent1、Component2、Component3の3つとすると、

Module=Component1 ; Component1.bas

Module=Component2 ; Component2.bas

Module=Component3 ; Component3.bas

といった内容を含むプロジェクトファイルが生成される。

【 0 0 8 4 】

(3-2) パラメータ定義生成手段 4 3 及びパラメータ定義生成ルール 8 2

図 1 5 にパラメータ定義生成手段 4 3 の処理例を示す。パラメータ定義生成手段 4 3 は、クライアントモジュール生成手段 4 2 からコンポーネント名を指定して呼び出されると、定義情報抽出手段 4 1 を通じてソフトウェアコンポーネント定義情報 5 0 から、当該コンポーネント名に対応するコンポーネント情報 5 2 配下の全メソッド情報 5 3 における全パラメータ名 5 4 1 とその型 5 4 2 とを取得し(S 6 1)、パラメータ定義生成ルール 8 2 に従って、パラメータ定義を生成する(S 6 2)。

【 0 0 8 5 】

パラメータ定義生成ルール 8 2 には、Dim の後にパラメータ名 5 4 1 とその型 5 4 2 をAsで連結してパラメータ定義を生成する旨のルールが用意されている。従って、パラメータ名 5 4 1 をParam1、Param2、Param3 の3つとし、その型 5 4 2 をString、Integer、Longとすると、以下のようなパラメータ定義が生成される。

Dim Param1 As String

```
Dim Param2 As Integer
```

```
Dim Param3 As Long
```

【 0 0 8 6 】

(3 - 3) 前処理生成手段 4 4 及び前処理生成ルール 8 3

図 1 6 に前処理生成手段 4 4 の処理例を示す。前処理生成手段 4 4 は、クライアントモジュール生成手段 4 2 からモジュール名及びコンポーネント名を指定して呼び出されると、コンポーネント宣言生成ルール 8 3 1 に従って、コンポーネントの宣言を生成する (S 7 1) 。

【 0 0 8 7 】

コンポーネント宣言生成ルール 8 3 1 には、例えば以下のようなルールが予め用意されている。

「以下のテンプレートのモジュール名とコンポーネント名の箇所に、モジュール名とコンポーネント名を設定して、コンポーネントの宣言を生成する。

```
Dim コンポーネント名 As Object
```

```
Set コンポーネント名=CreateObject( " モジュール名. コンポーネント名" )
```

【 0 0 8 8 】

従って、モジュール名をModule1、コンポーネント名をComponent1とすると、以下のようなコンポーネントの宣言が生成される。

```
Dim Component1 As Object
```

```
Set Component1=CreateObject( " Module1.Component1" )
```

【 0 0 8 9 】

次に前処理生成手段 4 4 は、当該モジュール名且つ当該コンポーネント名の配下にあるメソッド情報 5 3 の配下のパラメータ情報 5 4 を定義情報抽出手段 4 1 を通じてソフトウェアコンポーネント定義情報 5 0 から参照し、入力パラメータ設定処理生成ルール 8 3 2 に従って、入力となるパラメータ (入力パラメータ及び入出力パラメータ) の設定処理を生成する (S 7 2) 。具体的には、パラメータ情報 5 4 中の各パラメータ名 5 4 1 毎に、パラメータの入出力種別の情報 5 4 3 を参照して、入力パラメータ又は入出力パラメータとなるパラメータ名 5 4 1 だけを抽出し、入力パラメータ設定処理生成ルール 8 3 2 に従って、入力パラメー

タの設定処理を生成する。

【 0 0 9 0 】

入力パラメータ設定処理生成ルール 8 3 2 には、例えば以下のようなルールが予め用意されている。

「以下のテンプレートのパラメータ名の箇所にパラメータ名を設定して、入力パラメータの設定処理を生成する。

パラメータ名 = ?」

【 0 0 9 1 】

従って、Param1、Param2、Param3 が入力または入出力パラメータ名とすると、以下のような入力パラメータ設定処理が生成される。

Param1 = ?

Param2 = ?

Param3 = ?

【 0 0 9 2 】

(3 - 4) コンポーネント呼び出し生成手段 4 5 及びコンポーネント呼び出し生成ルール 8 4

図 1 7 にコンポーネント呼び出し生成手段 4 5 の処理例を示す。コンポーネント呼び出し生成手段 4 5 は、クライアントモジュール生成手段 4 2 からモジュール名およびコンポーネント名を指定して呼び出されると、定義情報抽出手段 4 1 を通じてソフトウェアコンポーネント定義情報 5 0 から、指定されたモジュール名かつコンポーネント名を持つコンポーネント情報 5 2 配下の全てのメソッド情報 5 3 中のメソッド名及びメソッドの呼び出し順の情報 5 3 4 と、各メソッド情報 5 3 配下の全てのパラメータ名 5 4 1 を取得し、コンポーネント呼び出し生成ルール 8 4 に従って、呼び出し順の情報 5 3 4 で指定された通りに各メソッドを呼び出すコンポーネント呼び出しを生成する (S 8 1) 。

【 0 0 9 3 】

コンポーネント呼び出し生成ルール 8 4 には、例えば以下のようなルールがメソッドの呼び出し形式の種類毎に予め用意されている。

「以下のテンプレートのコンポーネント名、メソッド名、パラメータ名の箇所に

それぞれの名前を設定して、コンポーネント呼び出し処理を生成する。

Call コンポーネント名. メソッド名 (パラメータ名) 」

【 0 0 9 4 】

従って、コンポーネント名をComponentName、メソッド名をBind、Send、Recv、UnBindの4つとし、SendのパラメータとしてParam1, Param2があり、RecvのパラメータとしてParam3、Param4があり、呼び出し順の情報534が指定するメソッドの呼び出し順が、Bind、Send、Recv、UnBindの順であったとすると、例えば以下のようなコンポーネント呼び出し処理が生成される。

Call ComponentName.Bind()

Call ComponentName.Send(Param1, Param2)

Call ComponentName.Recv(Param3, Param4)

Call ComponentName.UnBind()

因みに、Call ComponentName.Send(Param1, Param2)の直前には、Param1, Param2 の内で入力となるパラメータについて前処理生成手段44で生成された値の設定処理が、その直後には、Param1, Param2 の内で出力となるパラメータについて後処理生成手段46で生成された値の取得処理が組み込まれる。また、Call ComponentName.Recv(Param3, Param4)の直前には、Param3, Param4 の内で入力となるパラメータについて前処理生成手段44で生成された値の設定処理が、その直後には、Param3, Param4 の内で出力となるパラメータについて後処理生成手段46で生成された値の取得処理が組み込まれる。

【 0 0 9 5 】

(3-5) 後処理生成手段46及び後処理生成ルール85

図18に後処理生成手段46の処理例を示す。後処理生成手段46は、クライアントモジュール生成手段42からモジュール名及びコンポーネント名を指定して呼び出されると、当該モジュール名且つ当該コンポーネント名の配下にあるメソッド情報53の配下のパラメータ情報54を定義情報抽出手段41を通じて参照し、後処理生成ルール85中の出力パラメータ設定処理生成ルール(図示せず)に従って、出力となるパラメータ(出力パラメータ及び入出力パラメータ)からの値の取得処理を生成する(S91)。具体的には、パラメータ情報54中の

各パラメータ名 5 4 1 毎に、パラメータの入出力種別の情報 5 4 3 を参照して、出力パラメータ又は入出力パラメータとなるパラメータ名 5 4 1 だけを抽出し、出力パラメータ設定処理生成ルールに従って、出力パラメータの値の取得処理を生成する。

【 0 0 9 6 】

後処理生成ルール 8 5 中の出力パラメータ設定処理生成ルールには、例えば以下のようなルールが予め用意されている。

「以下のテンプレートのパラメータ名の箇所にパラメータ名を設定して、出力パラメータの値の取得処理を生成する。

? = パラメータ名」

【 0 0 9 7 】

従って、Param1、Param2、Param3 が出力または入出力パラメータ名とすると、以下のような入力パラメータ設定処理が生成される。

? = Param1

? = Param2

? = Param3

【 0 0 9 8 】

なお、別の実施例として、ソフトウェアコンポーネントを生成するのに必要な定義情報とクライアントプログラムを生成するのに必要な定義情報とを別々に持つ形態が考えられるが、そうすると、定義部が 2 つになり、ユーザによる定義も 2 回行わなければならない、生成の操作も別々に行わなければならない。これに対して前述の実施例では、クライアントプログラム自動生成部 4 0 とソフトウェアコンポーネント自動生成部 3 0 が共通の定義情報 5 0 を利用することで、定義部が単一のインタフェースを持てば良く、ユーザの操作も 1 回で済む利点がある。

【 0 0 9 9 】

【発明の効果】

以上説明したように本発明によれば、ソフトウェアコンポーネントを生成するのに必要な情報に加えて、メソッドの呼び出し順の情報など、ソフトウェアコン

ポーネントを利用するクライアントプログラムを生成するのに必要となる情報も含む定義情報を入力してソフトウェアコンポーネント定義情報として保存しておき、この定義情報に基づいてクライアントプログラムのソースコードを自動的に生成するようにしたので、メインフレームの知識を有しない者であっても、メインフレームにアクセスするソフトウェアコンポーネントを利用するクライアントプログラムを開発でき、典型的な部分などクライアントプログラムのソースコード中の殆どの部分が自動生成されていることと相まって、クライアントプログラムの開発者の負担を大幅に軽減することが可能である。

【図面の簡単な説明】

【図 1】

本発明の実施の形態にかかるソフトウェアコンポーネント自動生成システムのブロック図である。

【図 2】

本発明の実施例にかかるソフトウェアコンポーネント自動生成システムのブロック図である。

【図 3】

ソフトウェアコンポーネント定義情報の構成の概略を示す図である。

【図 4】

ソフトウェアコンポーネント定義部の構成例を示すブロック図である。

【図 5】

ソフトウェアコンポーネント定義情報中のモジュール情報、コンポーネント情報、メソッド情報及びパラメータ情報に含まれる情報の一部の例を示す図である。

【図 6】

ソフトウェアコンポーネント自動生成部の構成例を示すブロック図である。

【図 7】

ソフトウェアコンポーネント生成ルール中に含まれるルールの例を示す図である。

【図 8】

ソフトウェアコンポーネント自動生成部内のモジュール生成手段の処理例を示すフローチャートである。

【図 9】

ソフトウェアコンポーネント自動生成部内のコンポーネント生成手段の処理例を示すフローチャートである。

【図 1 0】

ソフトウェアコンポーネント自動生成部内のメソッド生成手段の処理例を示すフローチャートである。

【図 1 1】

ソフトウェアコンポーネント自動生成部内のパラメータ生成手段の処理例を示すフローチャートである。

【図 1 2】

クライアントプログラム自動生成部の構成例を示すブロック図である。

【図 1 3】

クライアントプログラム生成ルール中に含まれるルールの例を示す図である。

【図 1 4】

クライアントプログラム自動生成部内のクライアントモジュール生成手段の処理例を示すフローチャートである。

【図 1 5】

クライアントプログラム自動生成部内のパラメータ定義生成手段の処理例を示すフローチャートである。

【図 1 6】

クライアントプログラム自動生成部内の前処理生成手段の処理例を示すフローチャートである。

【図 1 7】

クライアントプログラム自動生成部内のコンポーネント呼び出し生成手段の処理例を示すフローチャートである。

【図 1 8】

クライアントプログラム自動生成部内の後処理生成手段の処理例を示すフロー

チャートである。

【図 1 9】

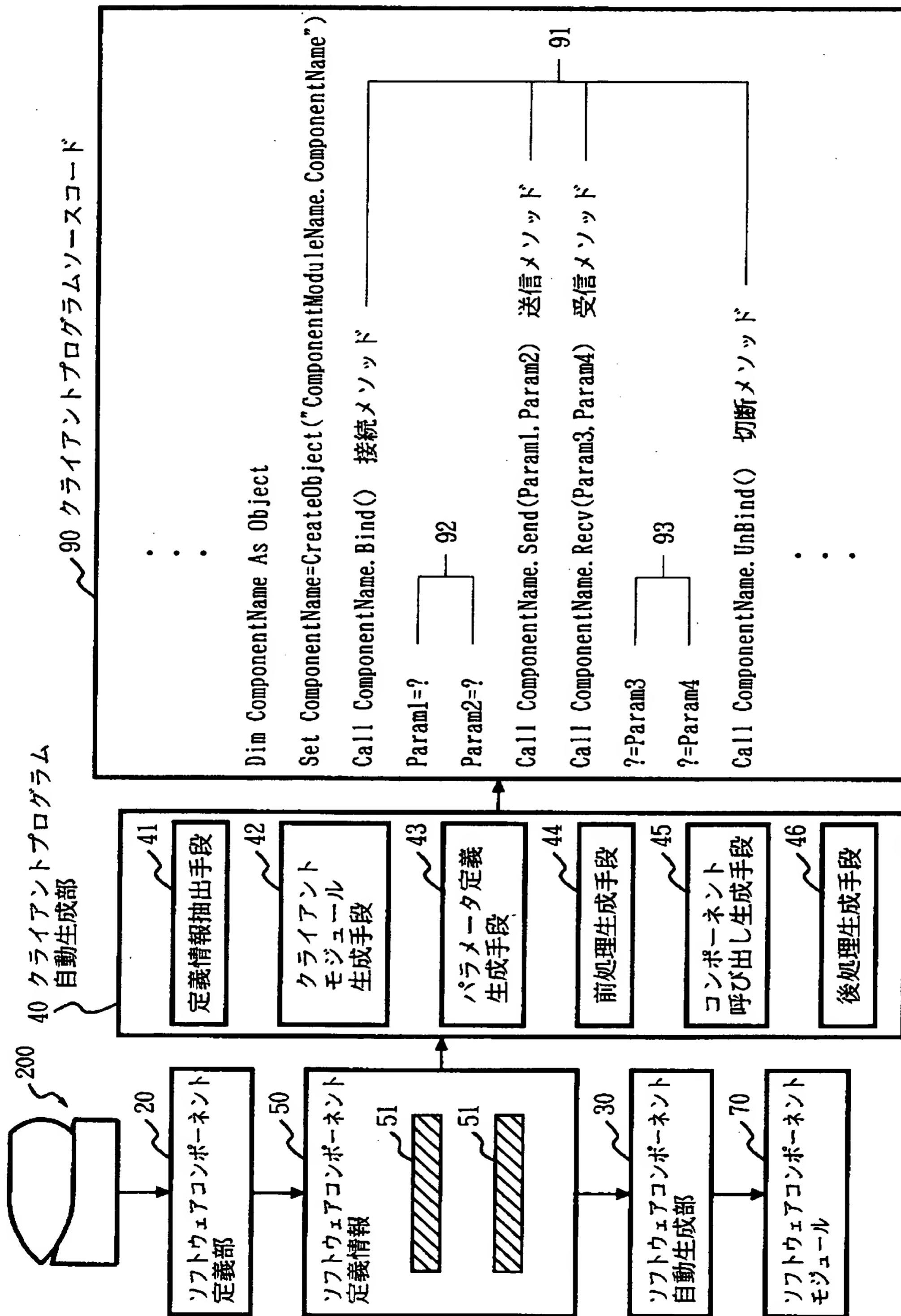
従来技術の説明図である。

【符号の説明】

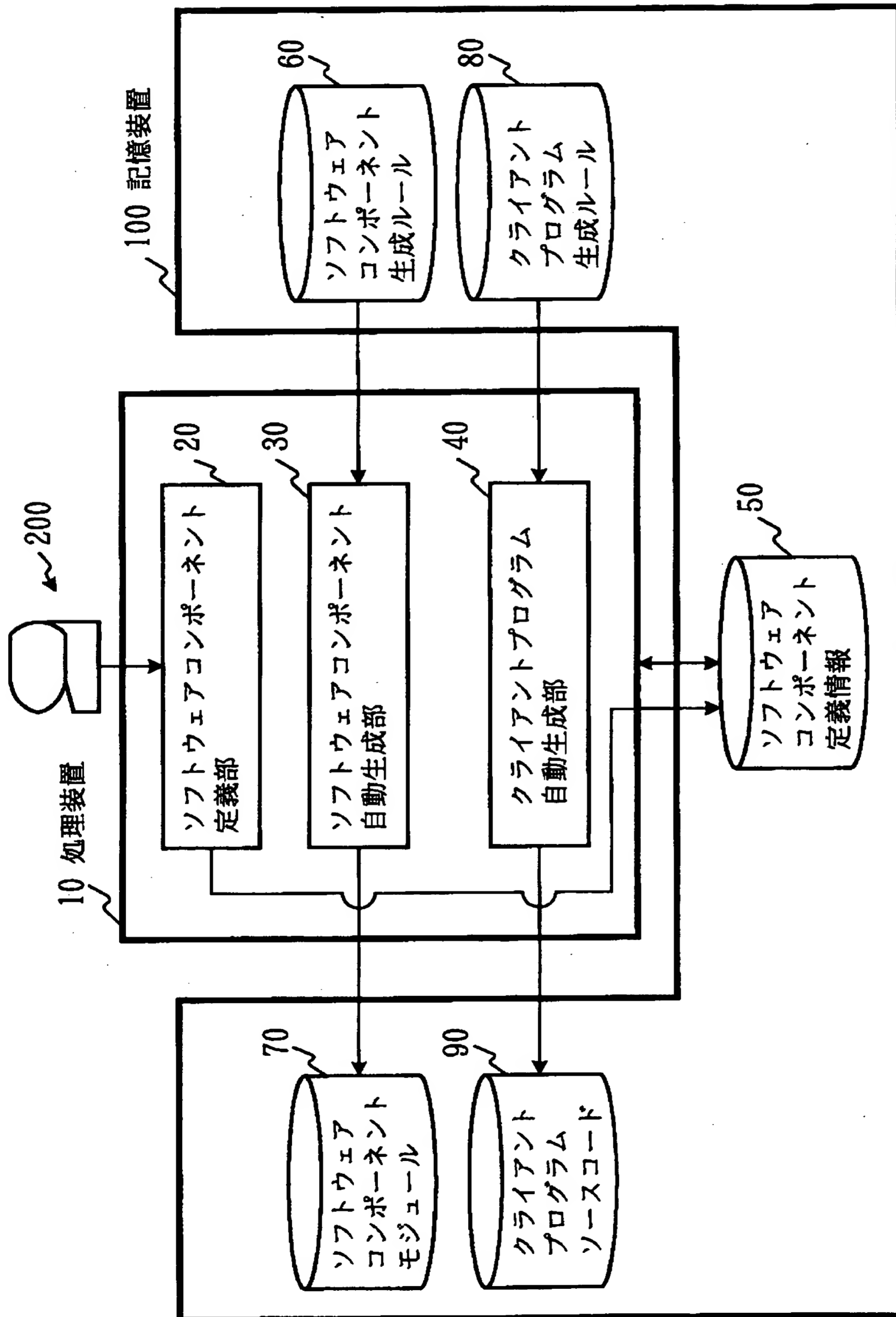
- 1 0 … 処理装置
- 2 0 … ソフトウェアコンポーネント定義部
- 3 0 … ソフトウェアコンポーネント自動生成部
- 4 0 … クライアントプログラム自動生成部
- 5 0 … ソフトウェアコンポーネント定義情報
- 6 0 … ソフトウェアコンポーネント生成ルール
- 7 0 … ソフトウェアコンポーネントモジュール
- 8 0 … クライアントプログラム生成ルール
- 9 0 … クライアントプログラムソースコード

【書類名】 図面

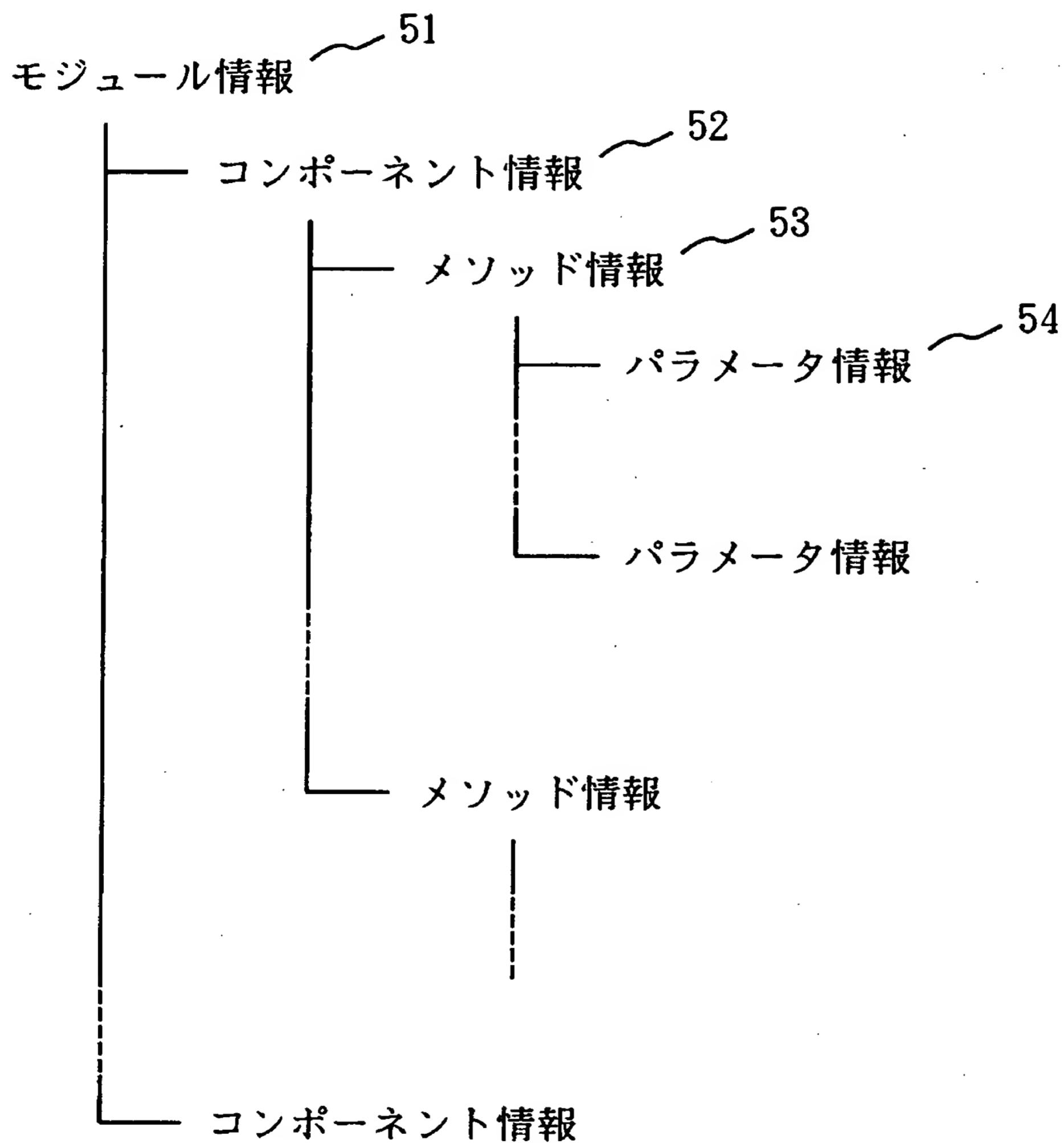
【図 1】



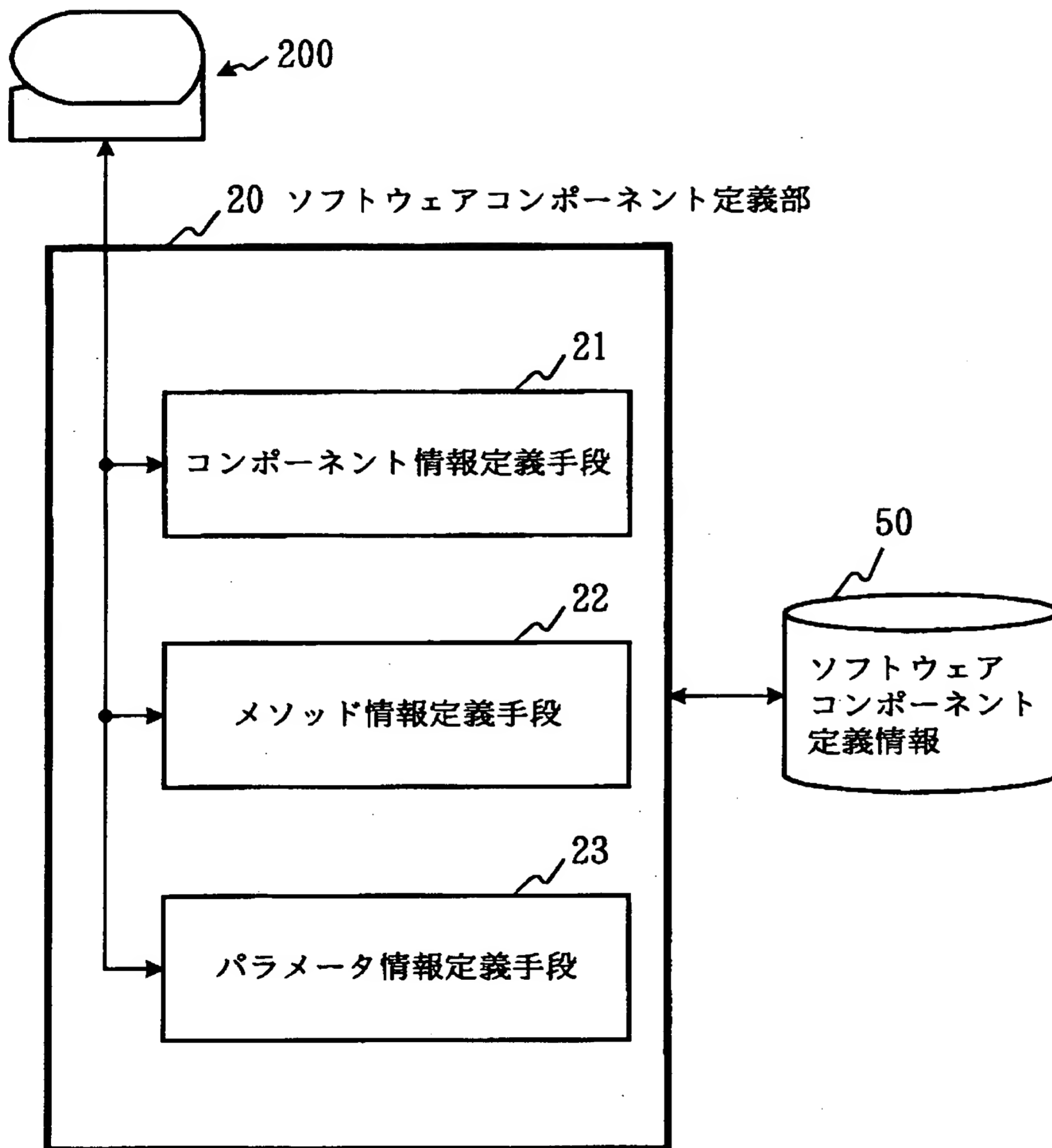
【図 2】



【図 3】



【図 4】



【図 5】

(a) モジュール情報 51

モジュール名・・・511

(b) コンポーネント情報 52

コンポーネント名・・・521

コンポーネント属性情報・・・522

(c) メソッド情報 53

メソッド名・・・531

メソッド属性情報・・・532

メソッドの呼び出し形式の情報・・・533

メソッドの呼び出し順の情報・・・534

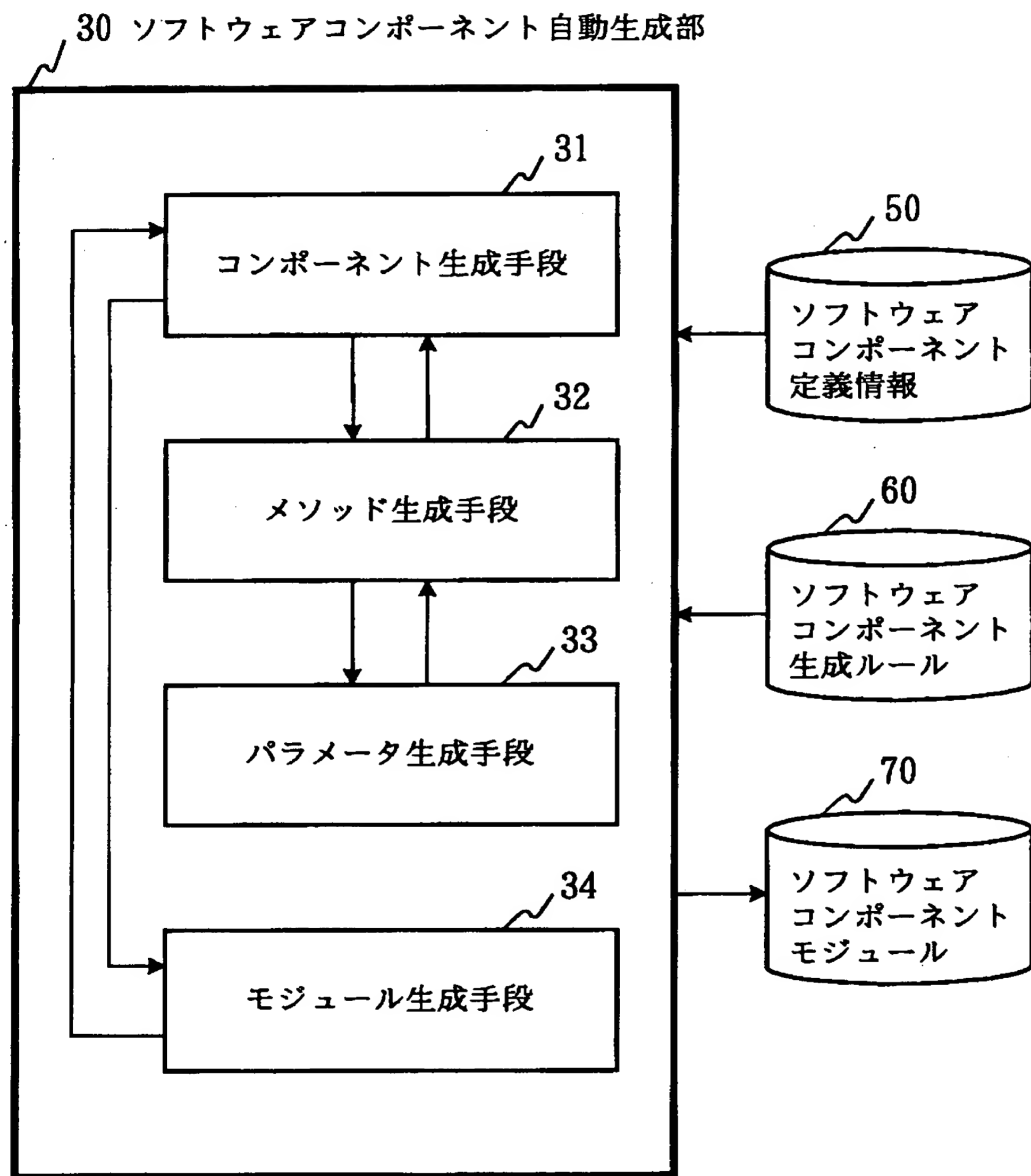
(d) パラメータ情報 54

パラメータ名・・・541

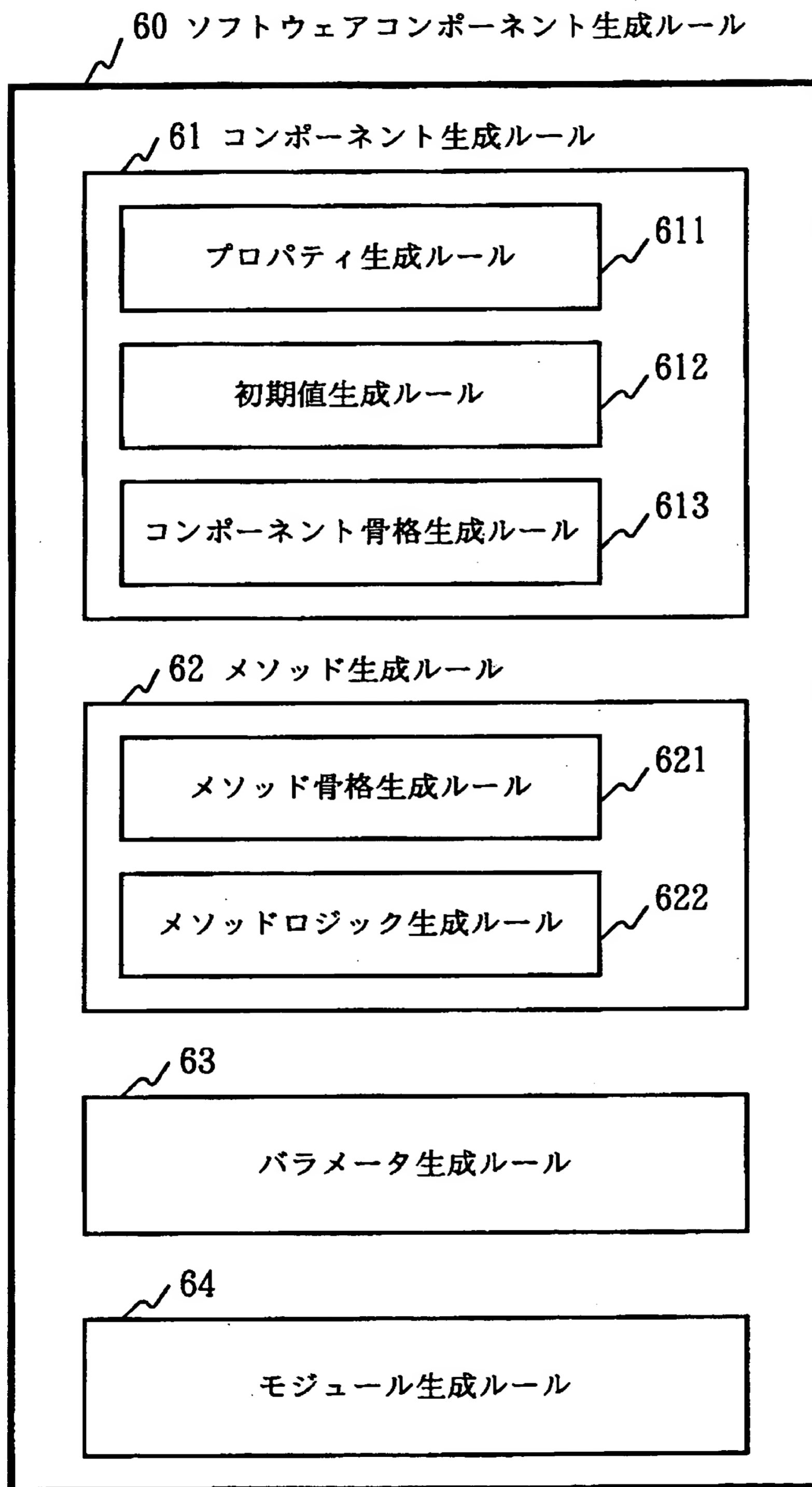
パラメータの型・・・542

パラメータの入出力種別の情報・・・543

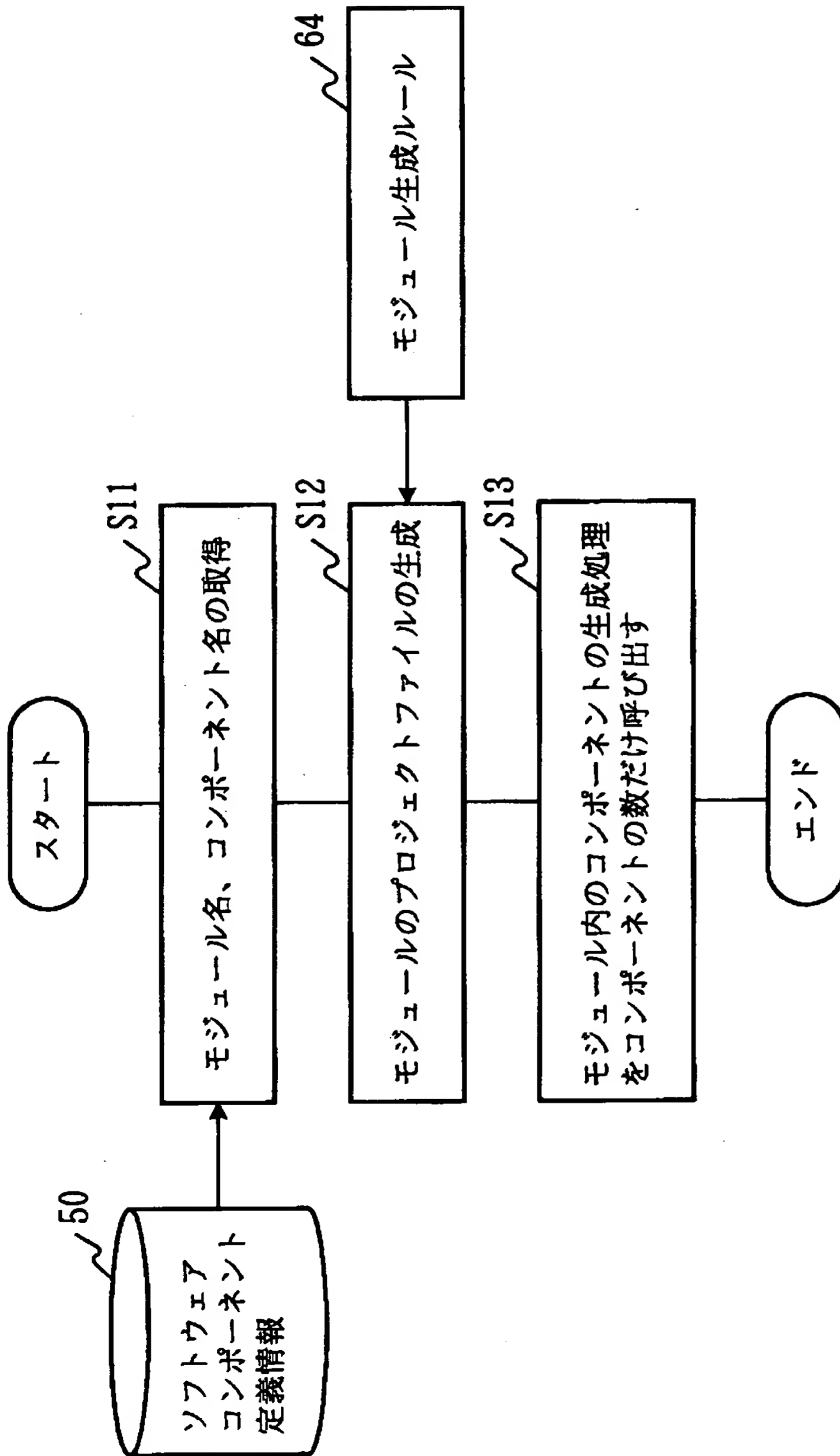
【図 6】



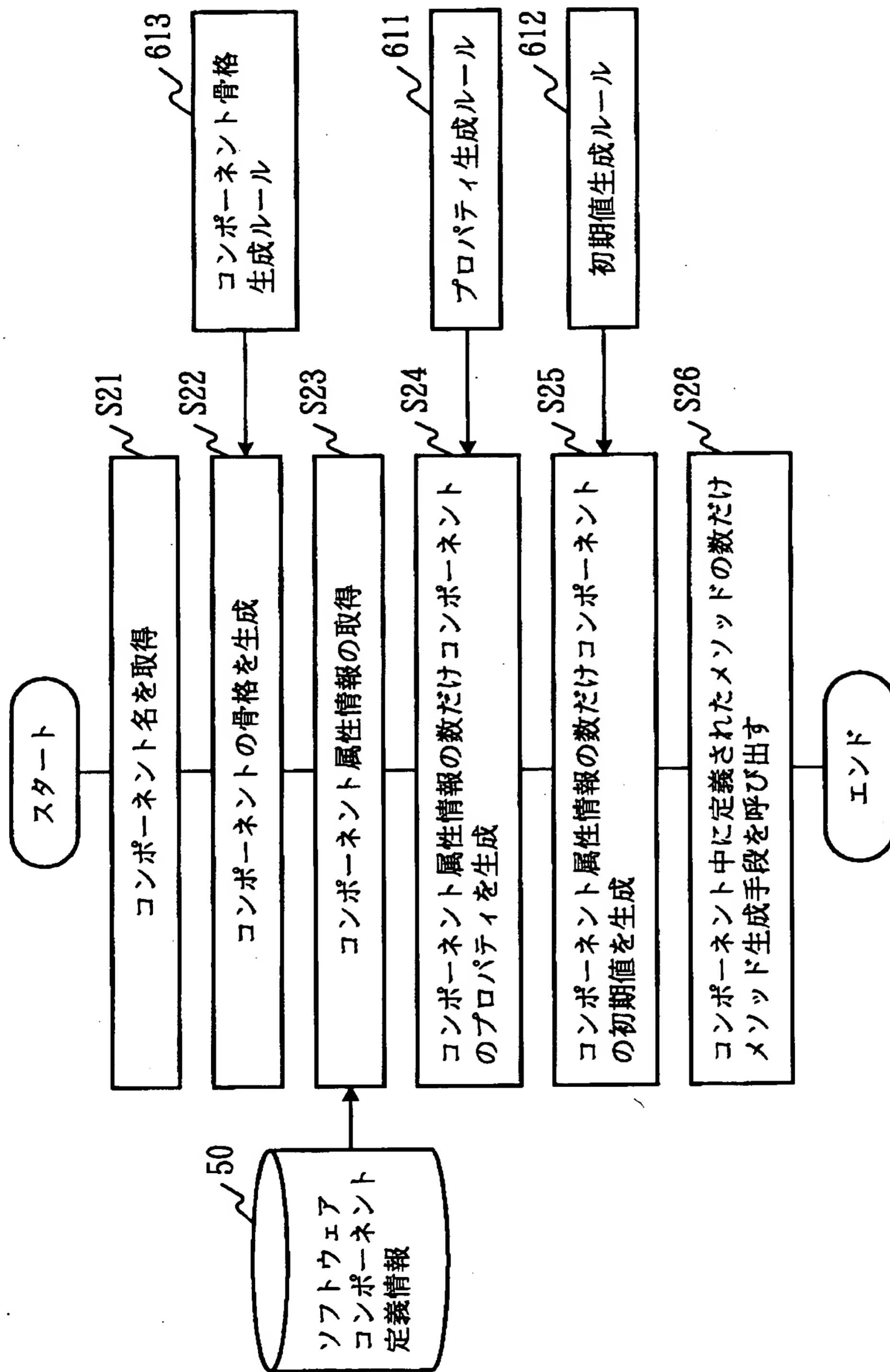
【図 7】



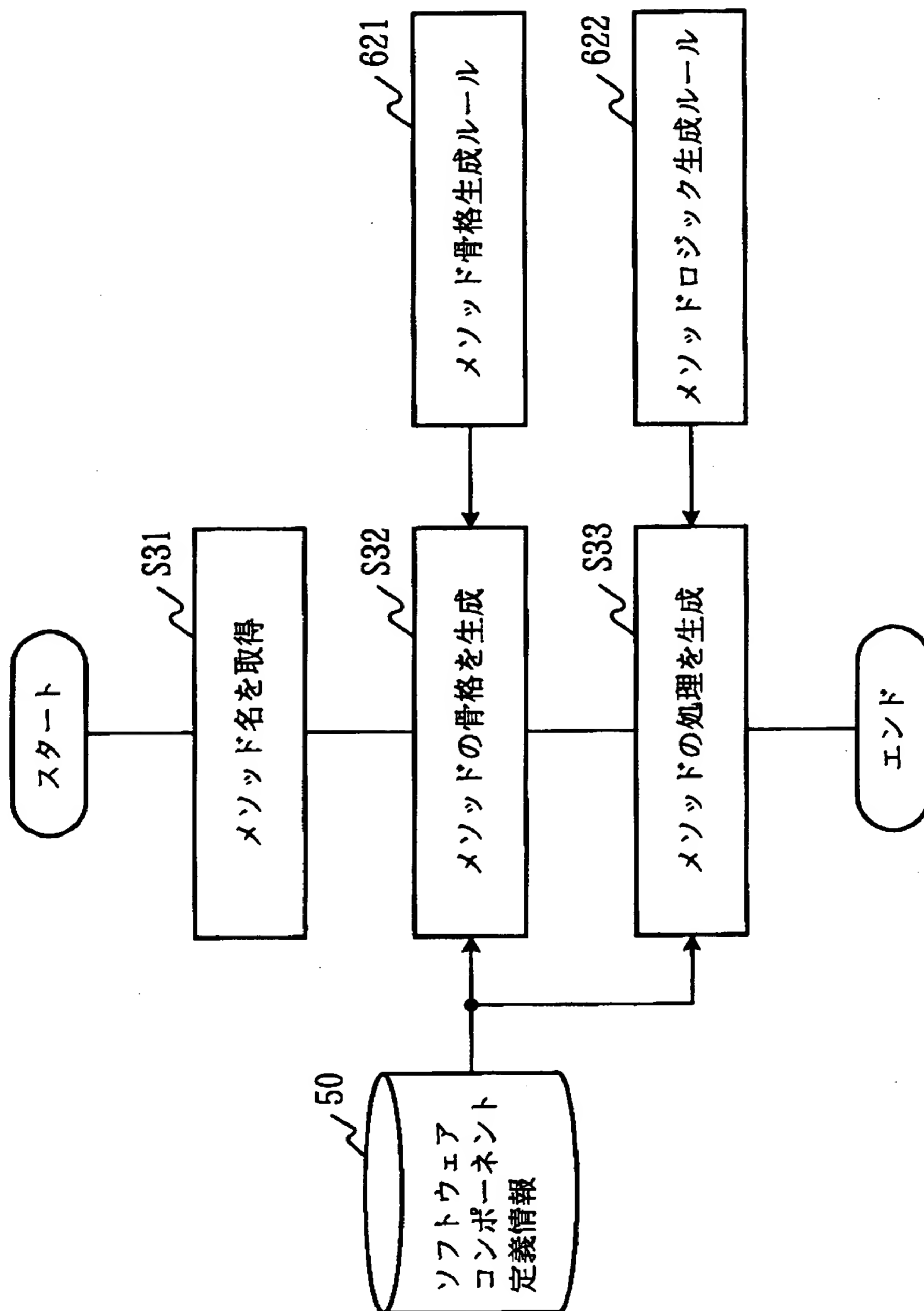
【図 8】



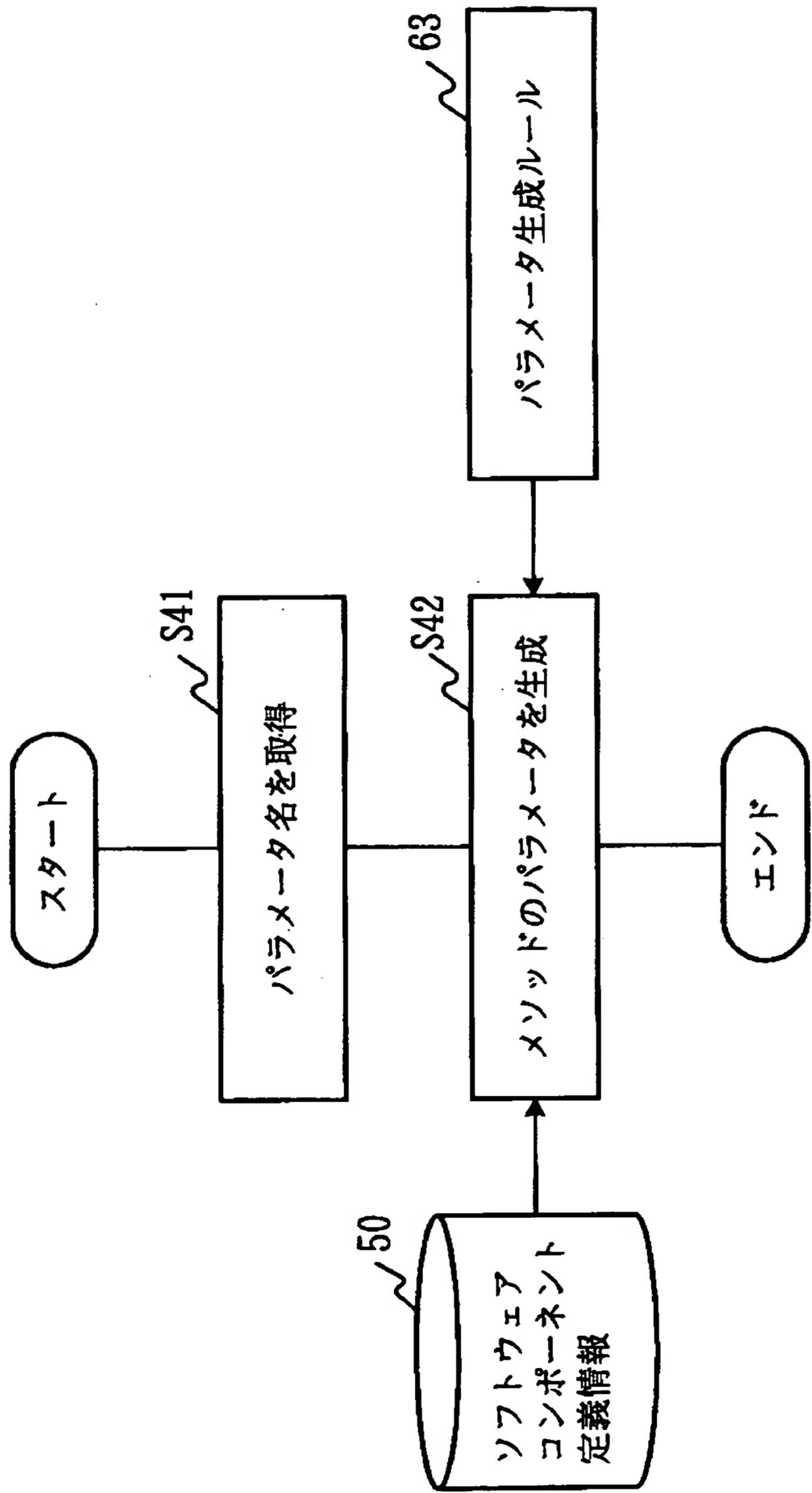
【図 9】



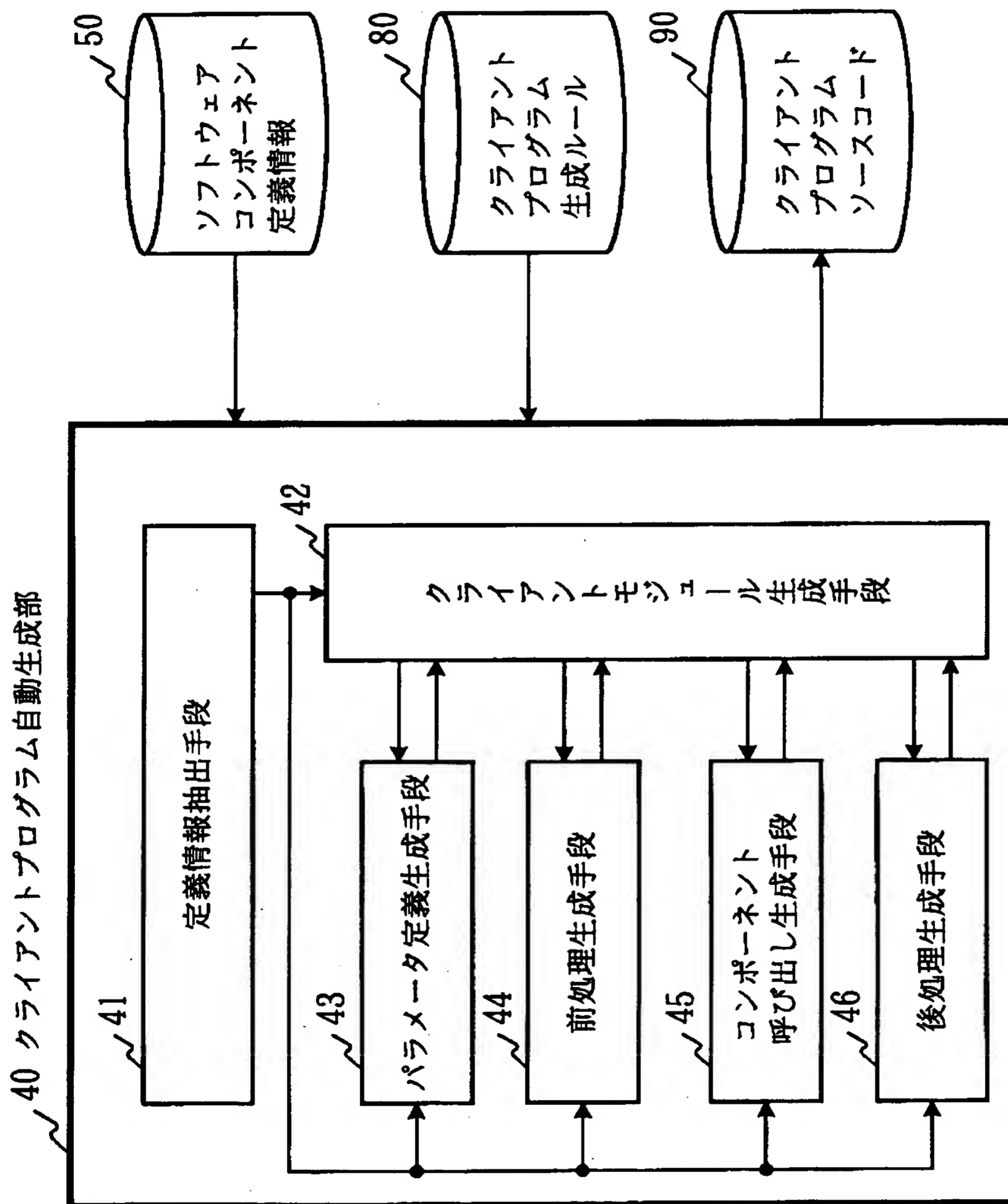
【図 1 0】



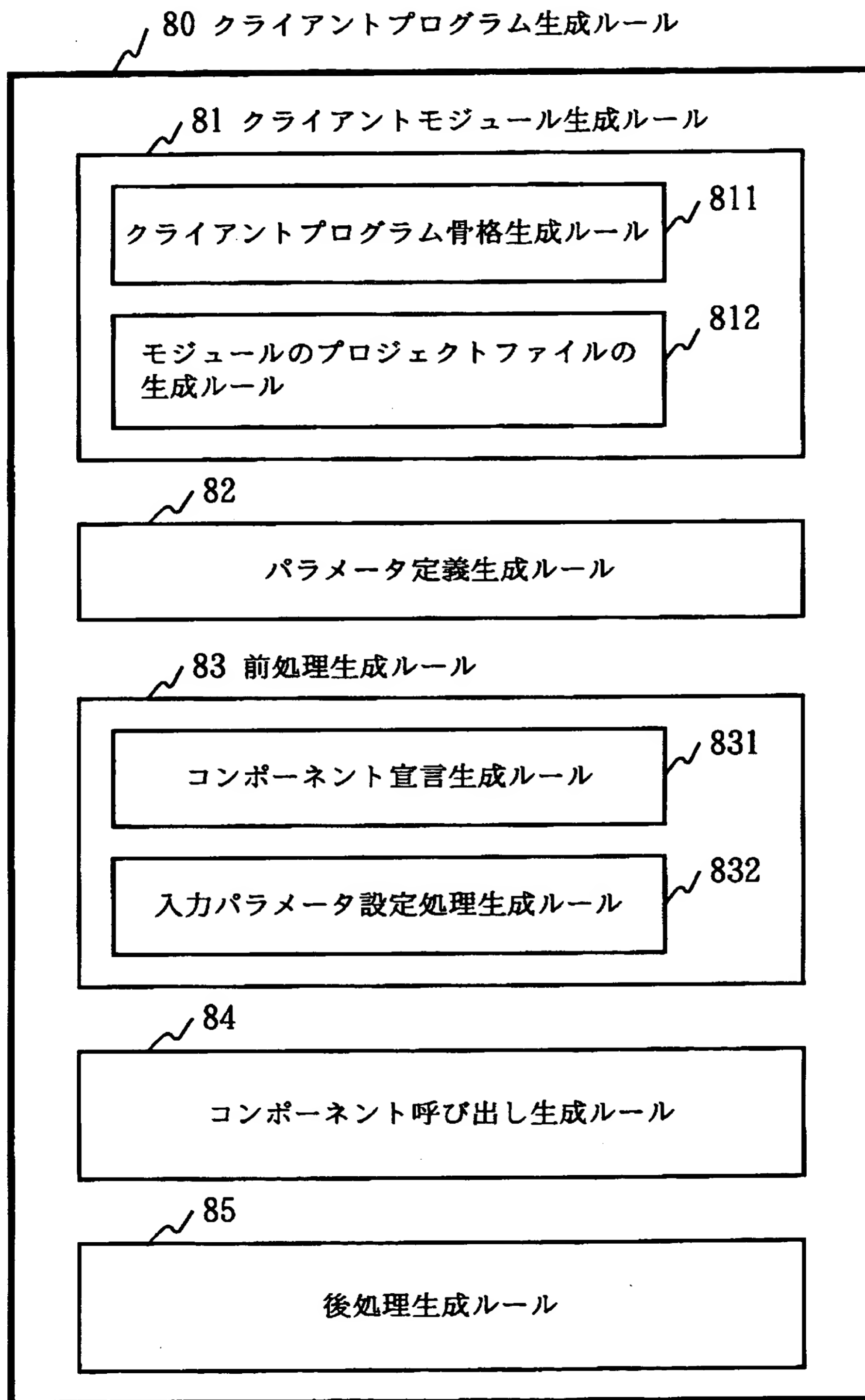
【図 1 1】



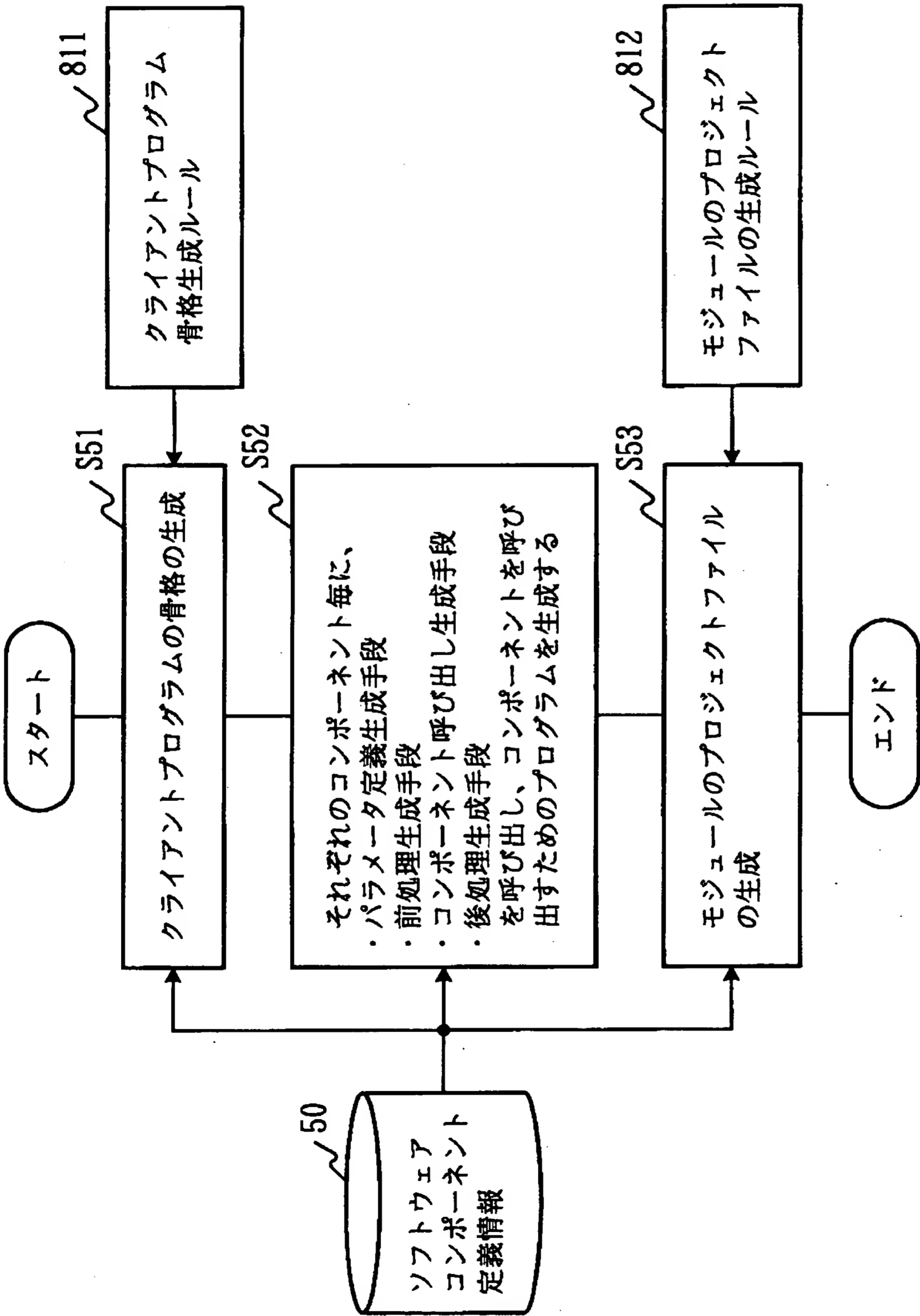
【図 12】



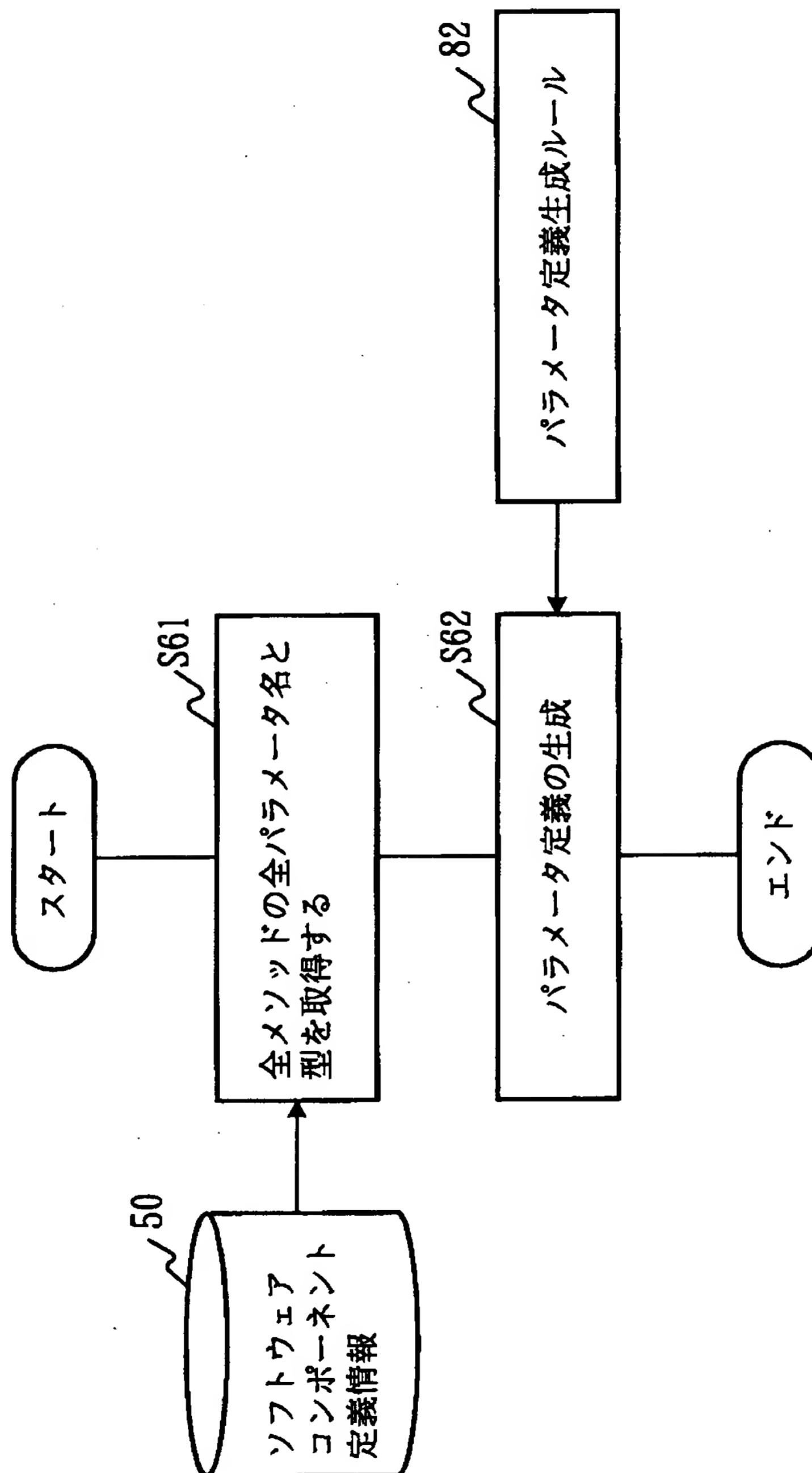
【図 13】



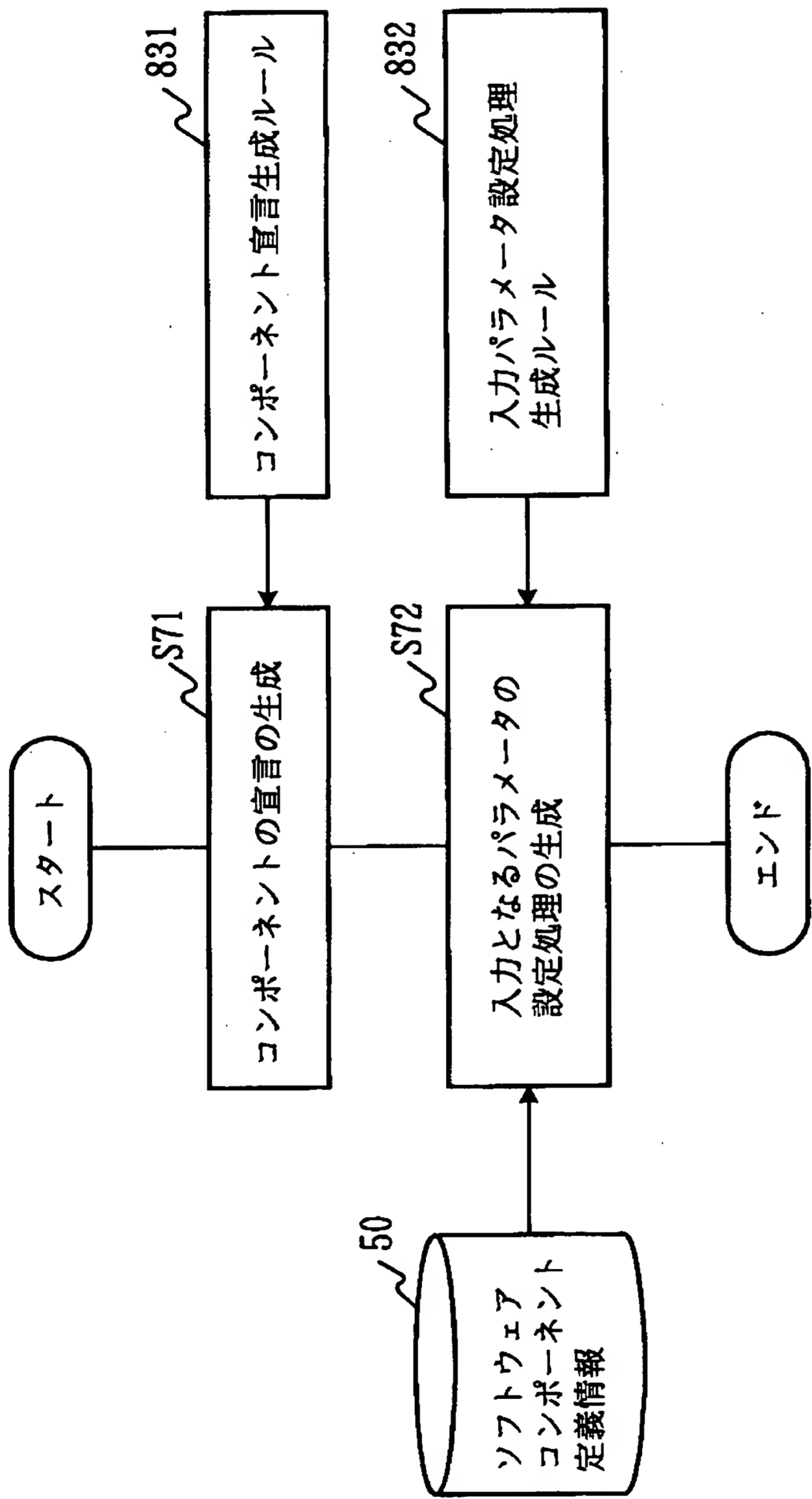
【図 1 4】



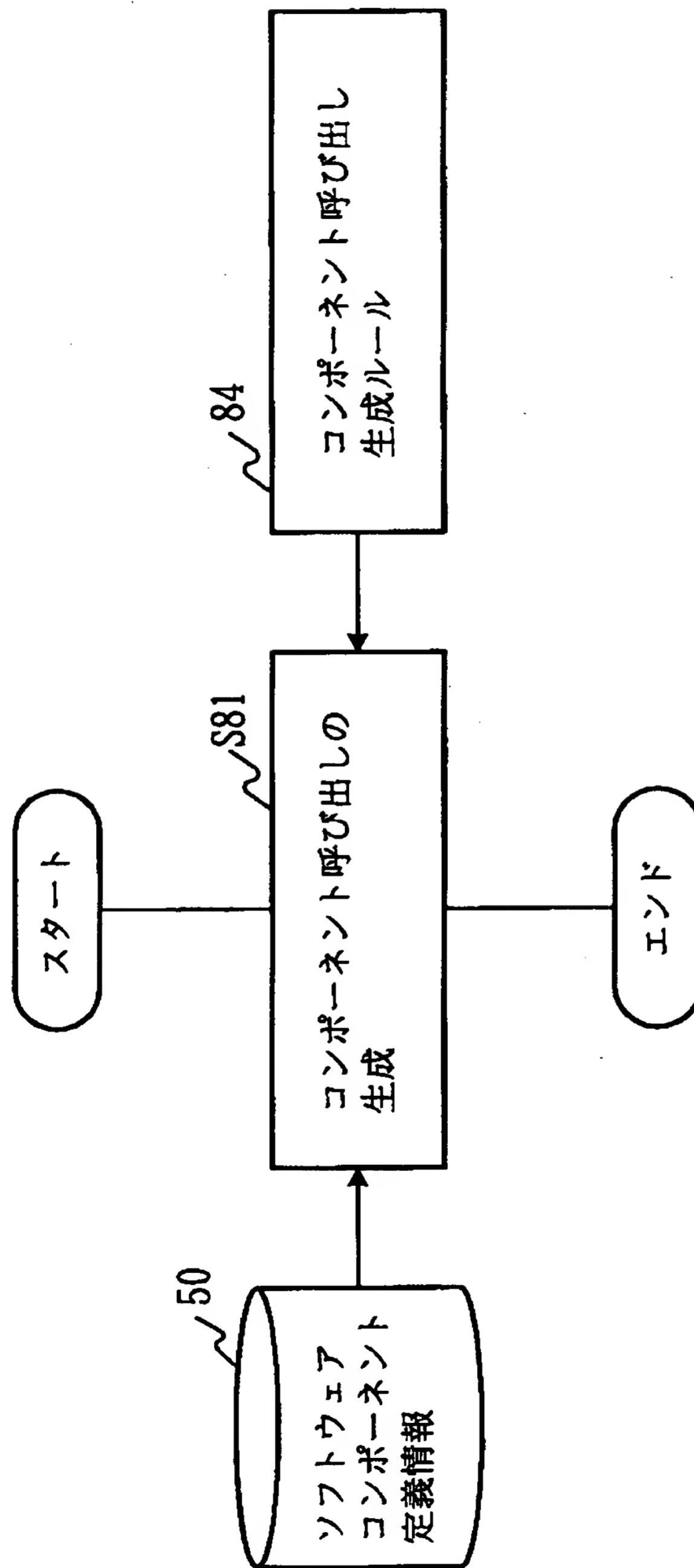
【図 1 5】



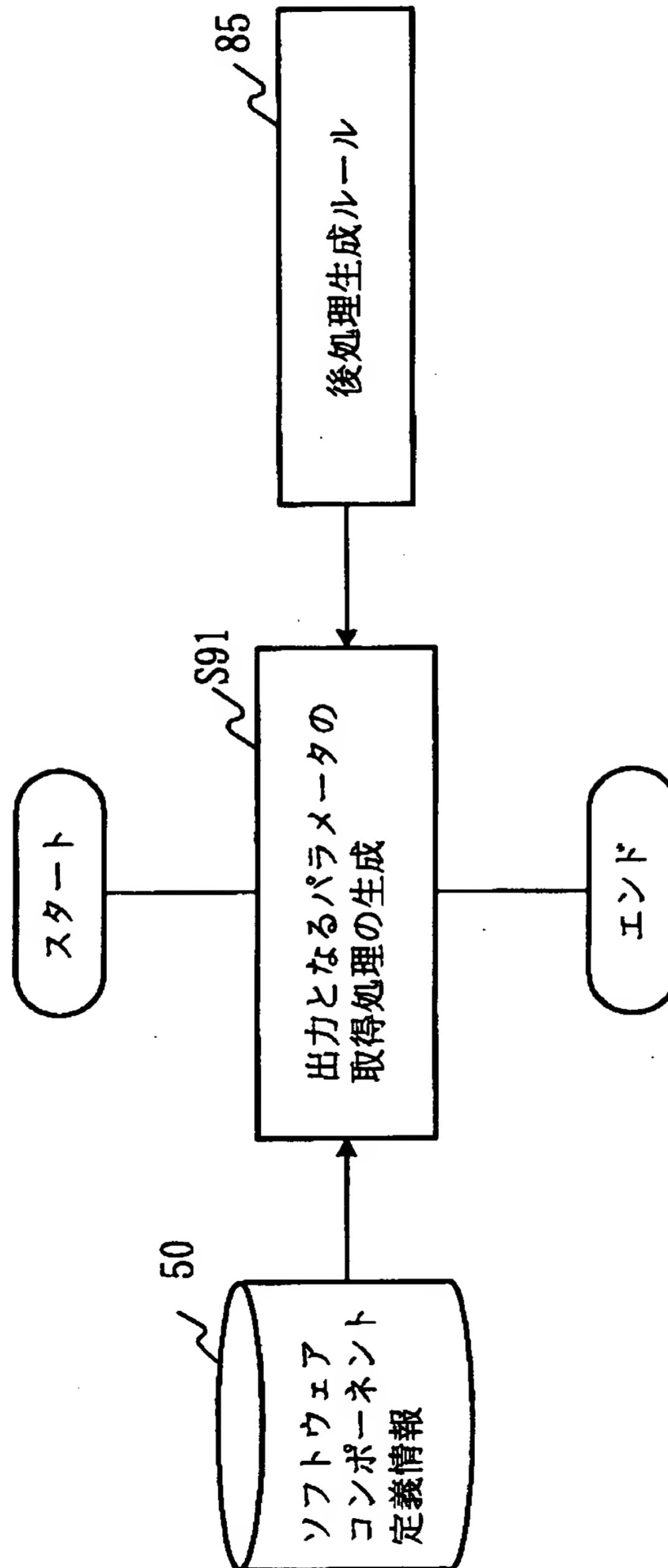
【図 1 6】



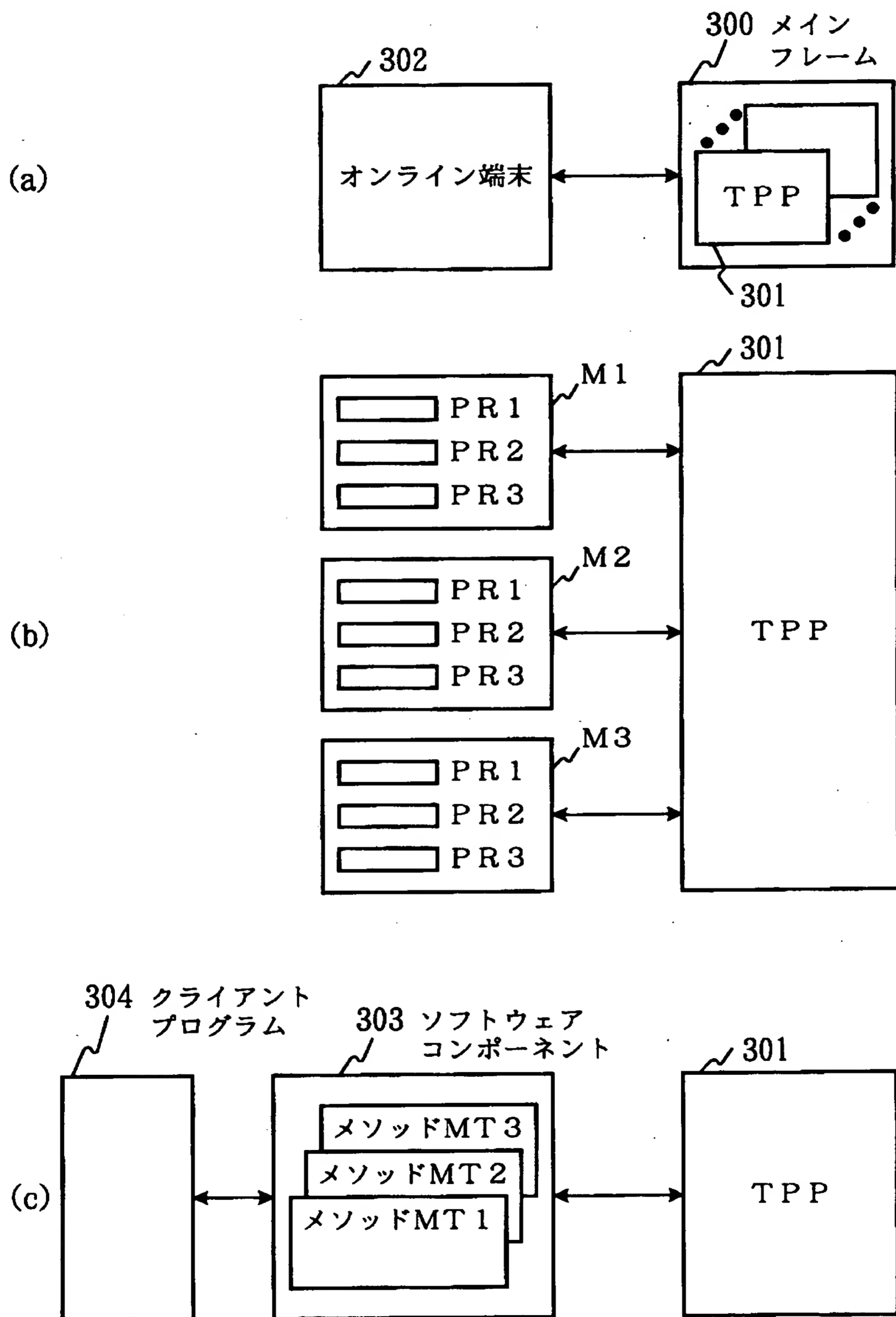
【図 1 7】



【図 1 8】



【図 1 9】



【書類名】 要約書

【要約】

【課題】 メインフレームにアクセスするソフトウェアコンポーネントを利用するクライアントプログラムの開発作業を支援するシステムを提供する。

【解決手段】 定義部20は、ソフトウェアコンポーネントの生成に必要な情報だけでなく、そのコンポーネントを利用するクライアントプログラムの生成に必要な情報（例えばメインフレーム上で動作するトランザクション処理プログラムに与えるメッセージと1対1に対応する前記コンポーネント中のメソッドの呼び出し順の情報など）も含む定義情報を利用者入出力装置200 から入力し、ソフトウェアコンポーネント定義情報50として保存する。クライアントプログラム自動生成部40は定義情報50に基づいてクライアントプログラムのソースコードのひな型を生成し、ソースコード90として出力する。ソースコード90に対して僅かな手作業を加えるだけで、クライアントプログラムを開発でき、開発者の負担が大幅に軽減される。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日 1990年 8月29日
[変更理由] 新規登録
住 所 東京都港区芝五丁目7番1号
氏 名 日本電気株式会社